2014-1602, -1603, -1604, -1605, -1606, -1607

IN THE
UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

PERSONAL WEB TECHNOLOGIES, LLC,
Appellant,

v.

EMC CORPORATION,
Appellee.

Appeals from the United States Patent and Trademark Office, Patent Trial and Appeal Board in Nos. IPR2013-00082, IPR2013-00083, IPR2013-00084, IPR2013-00085, IPR2013-00086, and IPR2013-00087.

CORRECTED BRIEF OF APPELLANT
PERSONALWEB TECHNOLOGIES, LLC

Roderick G. Dorman
Principal Counsel
Lawrence M. Hadley
McKOOL SMITH HENNIGAN, P.C.
865 South Figueroa Street, Suite 2900
Los Angeles, CA 90017
(213) 694-1200

Pierre J. Hubert
Joel L. Thollander
McKOOL SMITH, P.C.
300 W. 6th Street, Suite 1700
Austin, Texas 78701
(512) 692-8700

Daniel L. Geyser
McKOOL SMITH, P.C.
300 Crescent Court, Suite 1500
Dallas, TX 75201
(214) 978-4000

Attorneys for Appellant
PersonalWeb Technologies, LLC

November 12, 2014

CERTIFICATE OF INTEREST

Counsel for PersonalWeb Technologies, LLC certifies the following:

1. The full name of every party or amicus represented by me is:

   PersonalWeb Technologies, LLC

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

   N/A

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

   N/A

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

   McKool Smith Hennigan, P.C.: Roderick G. Dorman; Lawrence M. Hadley; Courtland L. Reichman

   McKool Smith, P.C.: Pierre J. Hubert; Joel L. Thollander; Daniel L. Geyser

   Nixon & Vanderhye: Joseph A. Rhoa; Updeep (Mickey) S. Gill

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES & RULES

STATEMENT OF RELATED CASES

Pursuant to FED. CIR. R. 47.5, Appellant PersonalWeb Technologies, LLC respectfully states that, aside from the six related Inter Partes Review proceedings consolidated in this single appeal:

(a) there have been no other appeals in or from the same proceedings in the lower tribunal before this or any other appellate court; and,

(b) the pending cases and proceedings that may be directly affected by this Court's decision in the pending appeal are as follows: *PersonalWeb Techs. LLC v. EMC Corp.*, No. 5-13-cv-1358 (N.D. Cal.); *PersonalWeb Techs. LLC v. Facebook Inc.*, No. 5-13-cv-1356 (N.D. Cal.); *PersonalWeb Techs. LLC v. NetApp, Inc.*, No. 5-13-cv-1359 (N.D. Cal.); *PersonalWeb Techs. LLC v. Google, Inc.*, No. 5-13-cv-1317 (N.D. Cal.); *PersonalWeb Techs. LLC v. Int'l Bus. Mach. Corp.*, No. 6-12-cv-661 (E.D. Tex.); *PersonalWeb Techs. LLC v. GitHub*, No. 6-12-cv-659 (E.D. Tex.); *PersonalWeb Techs. LLC v. Apple Inc.*, No. 6-12-cv-660 (N.D. Cal.); Patent Trial and Appellate Board, No. IPR2013-00596, U.S. Patent No. 7,802,310.

I. STATEMENT OF JURISDICTION

The Patent Trial and Appeal Board (PTAB) had jurisdiction over these Inter Partes Review (IPR) proceedings under 35 U.S.C. § 314, and issued its final written decisions on May 15, 2014. A36; A112; A186; A286; A388; A456. PersonalWeb timely filed its notices of appeal on May 20, 2014. A105; A157; A247; A366; A432; A513. This Court has jurisdiction over the consolidated appeals under 35 U.S.C. § 319 and 28 U.S.C. § 1295(a)(4)(A).

II. STATEMENT OF THE ISSUES

1. Whether the PTAB erred in construing three claim terms where: (a) for the means-plus-function term “identity means,” it devised its own structure instead of adopting the corresponding structure disclosed and linked in the specification; (b) for the means-plus-function term “existence means,” it devised its own function instead of adopting the function explicitly recited in the claim; and (c) for “sequence of non-overlapping parts,” it adopted a definition that covered disruptions to the sequence caused by intervening, non-sequential parts.

2. Whether the PTAB erred in holding the challenged claims anticipated where: (a) it repeatedly combined the elements of separate and distinct protocols described in prior-art references, rather than finding the elements arranged as in the claims; and (b) it repeatedly credited hypothetical embodiments of prior art-references, rather than analyzing the embodiments actually described.

3. Whether the PTAB erred in holding the challenged claims obvious where: (a) many of its conclusions were premised on faulty claim constructions or other legal mistakes; and (b) it failed to properly apply the *Graham* factors.

III. STATEMENT OF THE CASE

A. **Preliminary Statement.**

The decisions on appeal are "head scratchers." In its claim constructions, the PTAB repeatedly made legal determinations untethered to the teachings of the specification and the claims themselves—for one means-plus-function term, it crafted a structure not found in the specification; for another means-plus-function term, it crafted a function not found in the claims; and, for the term "sequence of non-overlapping parts," it crafted a definition that expressly covered non-sequential parts. Furthermore, two of these erroneous constructions were issued *after* the PTAB decided to institute these proceedings. It was as if the PTAB was stretching to support its initial determinations to grant the IPRs.

These surprising conclusions did not end with the claim constructions. They persisted in the PTAB's analyses of patentability. Only by impermissibly cobbling together disjointed elements in the prior art, conjuring up hypothetical embodiments found nowhere in any reference, and ignoring the technological limits of the embodiments actually disclosed in the prior-art references at issue could the PTAB reach its unpatentability determinations. The PTAB repeatedly acknowledged the rules governing its anticipation and obviousness inquiries, but then failed to apply them. Again, it was as if the PTAB was doing anything it could to justify and support its earlier determinations to institute these proceedings.

The new IPR procedures of the America Invents Act (AIA), for better or worse, incentivize the PTAB to become an advocate for its initial determination to grant an IPR. The old rule required a lower threshold determination; the new rule requires an initial determination of likelihood of success on the merits. Prior to the AIA, all that was needed to commence a reexamination was a determination that a "substantial new question of patentability" exists. Now the standard for instituting Inter Partes Review is whether "there is a reasonable likelihood that the petitioner will prevail with respect to at least one of the claims challenged in the petition." 35 U.S.C. § 314(a). In its decisions to institute these IPRs, the PTAB explains why it believes *it will* later adjudge the challenged claims to be unpatentable.

Human nature being what it is, it is difficult to persuade courts and panels to reverse themselves. Under this new AIA regime, the Federal Circuit alone must prevent human nature from trumping established law, and assure that valid patent claims are not wrongly held unpatentable.

B. **The True Name Patents Provide a Vital Solution for Identifying and Managing Data in Complex Computer Networks.**

The ability to reliably identify and locate specific data is essential to any computer system. On a single computer or within a small network, the task is relatively easy: simply name the data or file and identify it by that name and its stored location on the computer or within the network. Early operating systems facilitated this approach with standardized conventions for naming files, creating folder structures, and designating internal or attached storage devices, which together allowed the computer to locate the specific data. A2540(1:23-42). An example might look like this: c:\mydocuments\Budget_Forecast_1993.doc.

Ronald Lachman and David Farber recognized that conventional naming, locating, and managing schemes would be operationally inadequate as data processing systems continued expanding and new, distributed storage techniques were developed. A2540-41. As systems evolved, files could be divided and stored across different storage devices in dispersed geographic locations. While offering benefits, this also created a problem: different users could give identical names to different files or parts of files—or unknowingly give different names to identical files. Existing systems had no means to ensure that identical file names referred to the same data, and conversely, that different file names referred to different data. Lachman and Farber realized that, if these limitations were not surmounted, it would become infeasible to accurately identify, locate, retrieve, de-duplicate, replicate, and synchronize data within advanced systems. A2540-41.

Lachman and Farber had a solution: they developed a system that replaced conventional naming system-wide with "substantially unique," *content-based* identifiers. A2541(3:29-35). This approach could assign substantially unique identifiers to an endless variety of "data items"—"the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits." A2540(1:54-60). Applied system-wide, this invention would permit any data item to be stored, located, managed, synchronized, and accessed using its content-based identifier.

But how could a system generate a "substantially unique identifier"—based on content alone—for any size data item, system-wide? For this, Lachman and Farber turned to cryptography. Cryptographic hash functions, including MD4, MD5, and SHA, had been used in computer systems to verify the integrity of retrieved data—a so-called "checksum." A2546(13:15-19). Lachman and Farber recognized that these same hash functions could be devoted to a vital new purpose: if a cryptographic hash function was applied to a sequence of bits (a "data item"), it would produce a substantially unique result value, one that: (1) "virtually guarantee[s]" a different result value if the data item is changed; (2) is "computationally difficult" to reproduce with a different sequence of bits; and (3) cannot be used to recreate the original sequence of bits. A2546(13:3-8). These cryptographic hash functions would thus assign any sequence of bits—based on content alone—a substantially unique identifier. Lachman and Farber estimated that the odds of these hash functions producing the same identifier for two different sequences of bits (*i.e.*, the "probability of collision") would be at least 1 in 2^{29}. A2546(13:35-45). With such low probability of collision, Lachman and Farber dubbed their content-based identifier a "True Name." A2542.

With this insight, Lachman and Farber crafted novel ways for using True Names to manage the universe of data (each item correlated with a single True Name) in a network, no matter its complexity. They conceived various data structures, including a "Local Directory Extension Table" (124 LDE) and "True File Registry" (126 TFR), for systemically tracking and managing information about every data item, capturing each True Name and any user-provided name, location, and other information paired with that True Name:

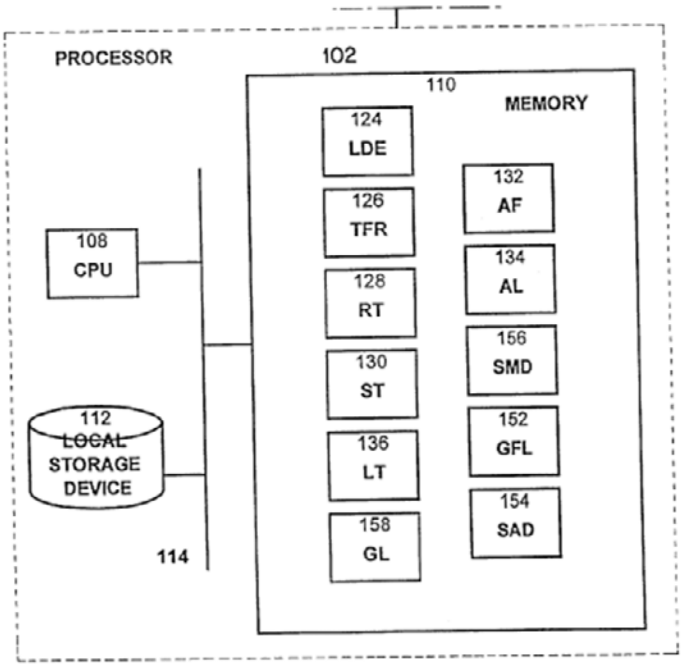


FIG. 1(b)

A2510; A2543(8:19-35). These data structures further permitted a key-map organization, allowing a rapid determination of whether any particular data item exists anywhere in a system and (if so) its location everywhere on that system. This essential functionality was simply not possible using the conventional art. A2510.

The invention envisions and allows for all data operations within the system to be managed using the True Name and associated information for each data item. This includes assimilating, identifying, and accessing all data items in the system by their True Name, regardless of actual storage location and user-designated conventional name. A2541. Several distinct advantages result—particularly in large, networked computer systems having multiple dispersed storage devices:

- With True Names, data no longer needs to be stored or transmitted as an indivisible unit. Files can be disassembled, segmented, and stored in different locations, and identical sets of data (or parts thereof) can be stored or transmitted as desired. Files can be segmented into random or fixed-length data items, each with a True Name independent of its location. Files (or file parts) can be identified by their constituent data items, each identified and accessed by a True Name, with the file later reassembled upon receipt. A2541(3:29-4:41).

- Duplicate data items can be eliminated or the amount of duplication can be optimized. This is useful particularly when files share identical “data item” parts, as each identical part will have the same True Name. The system can be configured to store only a fixed number of each data item (by True Name), thereby limiting duplication and optimizing storage space. A2541(3:48-53); A2550.

- Data items can be synchronized, replicated, and stored in geographically dispersed locations. Using a True Name, the system can determine whether a particular data item is present at a given location, and either copy or not copy to that location depending on whether the data item (or a predetermined number of copies) already exists. This allows for version management control, ensures that data items can be identified and accessed in the event that a particular storage device fails, and reduces bandwidth by allowing a user to identify the closest data item for retrieval. A2552(26:21-40); A2554.

On April 11, 1995, Lachman and Farber filed their patent application, describing these and other ways in which content-based "True Names" elevated data-processing systems over conventional file-naming systems. A2507. The first True Name patent—U.S. Pat. No. 5,978,791 (the '791 patent)—issued on November 2, 1999, A2507, followed by nine continuation patents, each claiming various techniques for using content-based True Name identifiers to rapidly access and efficiently store, manage, and transfer data.[1]

The True Name invention has been widely adopted. In particular, content-based naming has been employed and licensed in the various fields of cloud computing, backup systems, content-delivery networks, peer-to-peer networks, file-sharing applications, online streaming, search engines, and internet telephony. A23837-39; A24305-06.

[1] These consolidated appeals concern challenged claims drawn from six True Name patents: the '791 patent; U.S. Patent No. 6,415,280 (the '280 patent); U.S. Patent No. 7,945,544 (the '544 patent); U.S. Patent No. 7,945,539 (the '539 patent); U.S. Patent No. 7,949,662 (the '662 patent); and U.S. Patent No. 8,001,096 (the '096 patent). A36; A112; A186; A286; A388; A456.

C. The PTAB Focuses on Prior-Art References That Provide Different Solutions to Different Problems.

In declaring the claims at issue anticipated or obvious, the PTAB chiefly relied upon three references—a single "distributed storage" patent (Woodhill), A2823-49, an informal "file descriptions" newsgroup posting (Langer), A2570-75, and a user manual for a "contents signature" system (Kantor), A2576-613. For obviousness, the PTAB also combined these references with two unrelated sources—one "integrity check" patent (Fischer), A16763-75, and a limited set of "file system" articles (Satyanarayanan), A25466-78; A25961-73. These references, read alone or in combination, grappled with problems, and offered solutions, far different from those resolved by the True Name patents.

1. The Woodhill "distributed storage" patent.

The Woodhill patent concerns backup storage. A2823-49. It describes a system for backing up files in a computer network that has local work stations, each with a computer and storage disk. These local computers connect over a network to a remote backup file server:

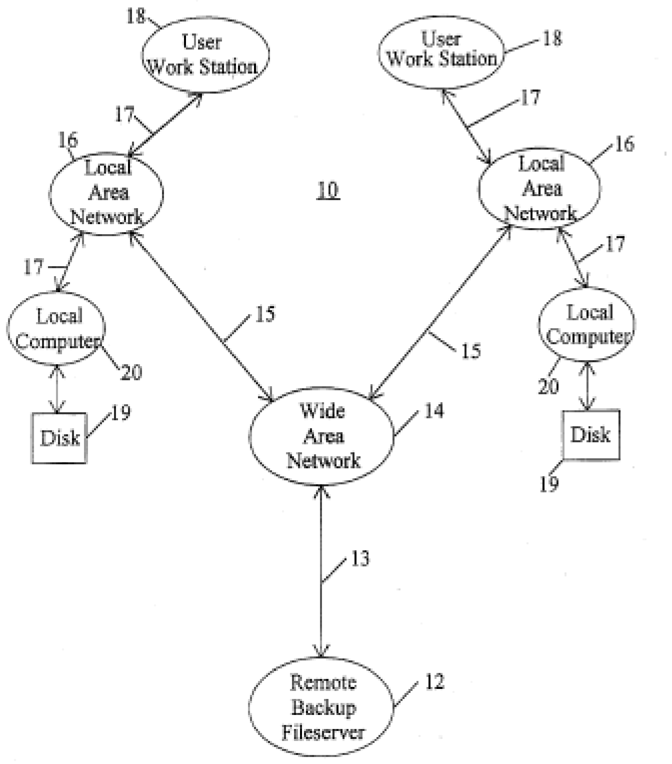


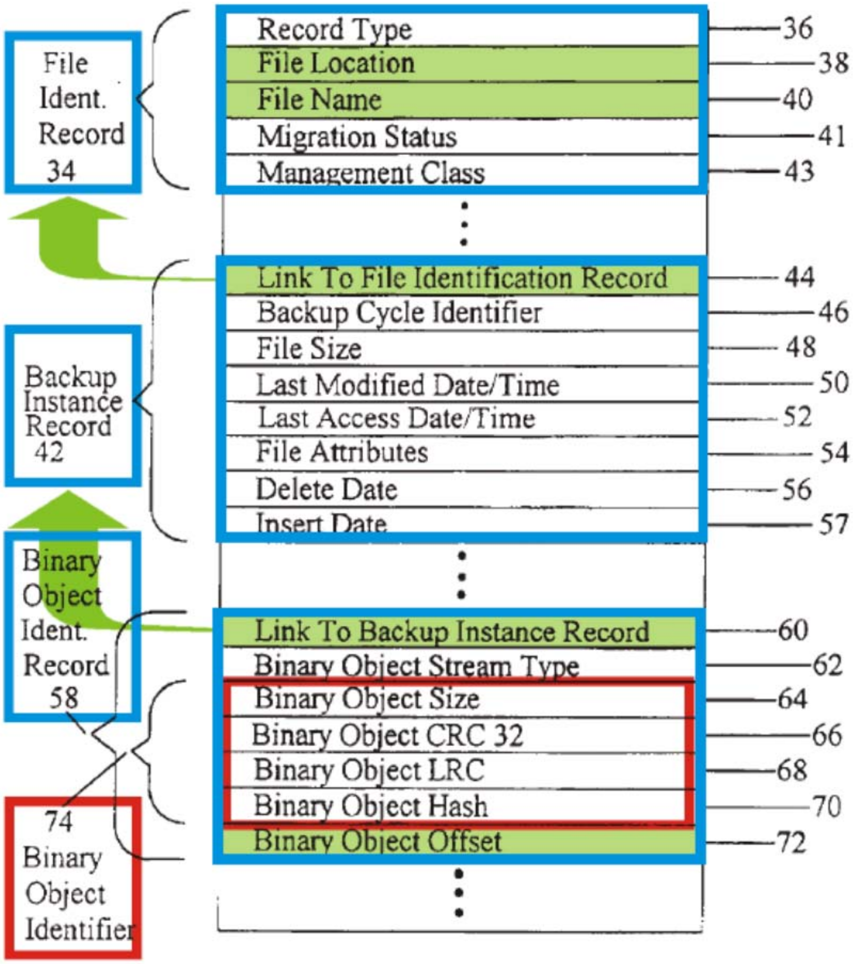
FIG. 1

A2824. Each local computer runs a Distributed Storage Manager program that allocates storage space and maintains a File Database for local and backed-up files. When a local computer backs up a file over 1MB (defined as a "convenient maximum binary object size") to the backup file server, the file is broken into "Binary Objects" of a 1MB fixed length—except for the final Binary Object, which may be under 1MB. A2839(4:23-25). Woodhill determines a content-based identifier for each Binary Object of a backed-up file (a "Binary Object Identifier"), and stores that identifier as part of a Binary Object Identification Record in the File Database on each local computer. A2841-42.

Woodhill's Binary Object Identifier is materially distinct from a True Name. For one thing, while the True Name algorithm is applied to "data items" of *any* size, the Woodhill algorithm is applied to "Binary Objects" of a fixed, 1 MB size (aside from the leftover segments or files under 1 MB). A2839. For another, Woodhill's Binary Object Identifier is not generated using a cryptographic hash. The Binary Object Identifier instead consists of four appended fields: (1) 32 bits representing the Binary Object size; (2) 32 bits from applying a cyclic redundancy check 32 (CRC-32) to the Binary Object; (3) 32 bits from applying a longitudinal redundancy check (LRC) to the Binary Object; and (4) 32 bits from applying an undisclosed "hash" algorithm to the Binary Object. A2841. This four-part identifier for Binary Objects is not a True Name cryptographic hash. A5934-35. It presumably suffices in Woodhill because its Binary Objects are relatively small (limited to a "convenient" maximum size), typically fixed-size, and need only uniquely identify Binary Objects for a single file (as opposed to providing unique identifiers for all files system-wide). A2841-42.

In Woodhill, a File Database contains a record for each file and its constituent Binary Objects. A2826. As depicted below, each record has a File Identification Record (34), a Backup Instance Record (42), and a Binary Object

Identification Record (58). Each Binary Object Identification Record contains a link (through the Backup Instance Record) to the file's user-defined subjective name (40) and its location (38). The Binary Object Identification Record also contains the four-part Binary Object Identifier fields, as well as a Binary Object Offset field, which determines each Binary Object within the parent file by counting bit offsets from the start of the file.



A2826 (annotating added).

Woodhill uses the Binary Object Identifier in two separate operations—a backup protocol and an audit protocol. A2840; A2846. Unlike True Names, these limited protocols do not use content-based identifiers to locate, identify, access, move, or synchronize all Binary Objects found anywhere on the system (within individual files or across different files). Woodhill's protocols use Binary Object Identifiers for two separate and straightforward uses: (1) to conduct one-to-one comparisons of Binary Objects within a known file at a known location during backup; and (2) to perform a conventional checksum audit.

1. *Backup Protocol*: Woodhill uses the Binary Object Identifier to determine whether part of a previously backed-up file should be replaced during a subsequent backup of the same file. A2840-42. In this operation, Woodhill initially determines, *without* using the Binary Object Identifier, whether a file on a local computer is new or has been previously backed-up and later modified. A2840. If the latter, Woodhill first recalculates a Binary Object Identifier for each Binary Object of the file. Next, Woodhill starts sequentially with the first Binary Object and compares the recalculated Binary Object Identifier against its counterpart in the File Database from the last backup. If the Binary Object Identifier has changed, then the Binary Object on the backup server is replaced with the latest version from the local drive. A2841. Using the Binary Object Stream Type and the Binary Object Offset field—neither parts of the Binary Object Identifier—Woodhill then

repeats the process with the next Binary Object. A2842(9:18-20). This way, only modified segments are replaced on the backup server.

Woodhill's backup process does not use the content-based Binary Object Identifier to determine if a local file has been previously backed-up or later modified. Woodhill uses the Backup Queue Record to determine if a file already exists on the backup server. For existing files, Woodhill determines whether a file has been modified by comparing other information in the Backup Queue Record with information in the file's file block (*e.g.*, user-defined file name, file location, and last-modified date and time). A2840(5:49-6:32). The True Names invention adopts the opposite approach: it uses the content-based True Name, not a user-defined name, to determine if a file exists at a storage location and whether it matches another file stored at a different location.

Woodhill's backup process is also limited to one-to-one comparisons: it compares, once, a single Binary Object Identifier with another single Binary Object Identifier. A2842. Unlike True Names, Woodhill never compares a Binary Object Identifier—for a Binary Object at a particular location in a particular file—against multiple Binary Object Identifiers for *other* Binary Objects at *all* locations in *all* files. For this reason, Woodhill can *only* determine whether a Binary Object exists in the *particular file*, and at the *particular offset*, being examined. In contrast, True Names operates system-wide: by comparing the substantially unique identifier against the identifiers for all data items, the True Name invention can determine whether a data item exists *anywhere in the system*. A2556-57.

These differences between Woodhill and True Names materially affect backup performance. Woodhill cannot limit the number of duplicate Binary Objects in separate files or different storage devices, or even avoid duplicates within the same file. Indeed, a single file can contain an unlimited number of identical Binary Objects—and Woodhill can do nothing about it. Lachman and Farber solved this critical problem in the True Name invention by comparing a data item's True Name against all True Names across a system or within a file, thereby permitting the number of duplicate, backed-up data items to be optimized (or even eliminated altogether). A2556-57.

These contrasting approaches reflect the stark difference between the object and scope of each invention. Woodhill focuses on backup functions; True Names focuses on *system-wide* data management. A2840; A2541. Because True Names operates system-wide, it can save bandwidth and storage space by optimizing the number and location of data items within complex systems (even those spread across distant locations). Rather than repeatedly recopying and restoring the same file, True Names can manage duplicates with the True File Registry. A2555-58. Woodhill, on the other hand, cannot control bandwidth or storage space even within its limited backup system. A2841.

2. *Audit Protocol*: Woodhill also describes a separate "auditing" process, which uses Binary Object Identifiers to perform integrity checks, much like the long-used "checksum" verification process. A2846. In this operation, Binary Objects are randomly selected from the remote backup server through the Binary Object Identification Record in the File Database. As the Binary Object is restored to a local computer, a new Binary Object Identifier is calculated for the Binary Object and compared with the Binary Object Identifier calculated when the file was last backed up. If the values are equal, a successful audit is recorded. If the values are not equal, an audit failure is recorded. A2846.

Woodhill does not use the Binary Object Identifier to *access* Binary Objects during the audit. Instead, Woodhill uses the *Binary Object Identification Record*, which contains a link to the file's name and location, and a field designating each Binary Object's offset from the first bit in the file. A2846(18:17-28). Woodhill's audit procedure thus accesses files conventionally, without any content-based identifier.[2] A2846(18:17-23); A2835; A2826.

[2] Indeed, using a file name, location, and Binary Object offset (and not the Binary Object Identifier) is the exact process, described elsewhere in Woodhill, used to access Binary Objects for other operations, including the access of a current version of a Binary Object so it can be restored to its previous version, A2846(17:18-36), as well as the access of Binary Objects for compression during the backup routine. A2843(11:57-12:5).

Notwithstanding these material differences between Woodhill and the True Name invention, the PTAB held that Woodhill anticipated claims 1-4, 29-33, and 41 of the ’791 patent, A40, claims 36 and 38 of the ’280 patent, A118, and claim 1 of the ’544 patent, A192, and rendered obvious claims 1-4 and 29 of the ’791 patent, A40, and claims 36 and 38 of the ’280 patent, A118.

2. The Langer “file descriptions” newsgroup posting.

Using an Australian e-mail address, Langer posted online comments to two newsgroups about an article discussing improvements to File Transfer Protocol (FTP) systems. A2570-75. Langer’s comments suggested further improvements to FTP systems—in particular speculating that future systems might benefit from a method of “uniquely identifying files” with different names, in different directories, or on different systems. Langer then described four ideas for using “unique identifiers” to improve FTP systems. None resembles the use of substantially unique identifiers in the True Name patents. A2572-73.

First, Langer notes that requesting a file posting by location, path, and filename will extract the file from the specified location even if the same file exists at a closer site. Langer suggests that users could be informed of the closest site containing a requested posting if an undefined *site* “identifier” were automatically inserted into all newsgroup postings. A2573. This idea is not analogous to the True

Names invention, which correlates content-based identifiers with data items, but never inserts *site* (or any other) identifiers into data items for any purpose.

Second, Langer suggests applying a hash function (such as MD5) to a file's contents so newsgroup users could perform conventional checksums when downloading. A2573. This was a well-known and conventional use of hash functions—one not claimed by the True Name patents.

Third, Langer suggests adding MD5 codes to the bibliographic information in file catalogues on the central servers. According to Langer, these codes could prevent sabotage by duplication of CRC codes, combat dissemination of viruses, and serve authentication functions. A2573. None of these uses are analogous to the system-wide use of cryptographic hashes in the True Name patents.

Fourth, Langer suggests "hardlink[ing] every file available for ftp to a filename encoding of it[s] MD5 token." Langer states that the "hardlink" would allow users to obtain a file by directory path and filename of the MD5 token. A2573. This proposed method is unlike the True Name invention, in which data items are accessed by comparing the True Name against a list of stored True Names, thus *eliminating* the need for directory path and filename information.

Notwithstanding these substantial differences between the Langer newsgroup posting and the True Name invention, the PTAB held that Langer anticipated claims 10 and 21 of the ’539 patent, and rendered obvious (in combination with Woodhill) claim 34 of the same patent. A292.

3. The Kantor “contents signature” user manual.

The Kantor user manual describes a software application designed to run on early electronic bulletin board systems. A2576-613. The software allows a board’s operator to check for duplicates among existing files, newly uploaded files, and files available through networks from other systems. This check solved a problem unique to electronic bulletin boards: to receive time credit, users would download a file, change its name, and re-upload the same file with the new name. With the adoption of compressed library files (such as zip files), this behavior began to burden systems and had the potential to cause serious harm. A2580-81.

The software described in Kantor creates a type of “contents signature” independent of the file’s name and date. To detect duplicates, the manual describes calculating a CRC-32 on the data for each file (including compressed zip files) in the bulletin board system. A2604. By comparing a CRC-32 (plus file length) on an uploaded file against the CRC-32 for existing files, the system could assume that a match flagged a duplicate file. A2580-81; A2604.

This rudimentary system differs markedly from the True Name invention. Unlike cryptographic hash functions, a CRC-32 does not produce a substantially unique identifier. A CRC-32 instead generates a result value that, depending on the volume of files in a system, will inevitably produce "collisions." A13941-43. When applied to different files, including in particular compressed files, a CRC-32 algorithm can yield the same result value—with increasing likelihood as bulletin boards accumulate a large volume of files. Accordingly, unlike a True Name, changing one or more bits in a file (particularly in a compressed file) does not necessarily result in a different CRC-32 value. A2602-04. Given this limitation, Kantor does not propose or describe (in contrast with True Names) substituting the CRC-32 bit "contents signature" for the file name. Nor does Kantor propose or describe (in contrast with True Names) using that "contents signature" for functions such as locating, identifying, accessing, synchronizing, or moving files or data items from one location to another. A2580-81.

Notwithstanding these material differences between the Kantor user manual and the True Name invention, the PTAB held that Kantor anticipated and rendered obvious (in combination with Woodhill) claim 1 of the '544 patent. A192. The PTAB further held that Kantor rendered obvious claims 10 and 21 of the '539 patent, as well as (in combination with Langer) claim 34 of the same patent. A292.

4. The Fischer "integrity check" patent.

The Fischer patent describes a method of calculating a conventional integrity check for files that undergoing continuous change, such as large database files. A16763-75. When new records are added to a database, a hash function is applied to the new record, and then a second commutative function (such as an "XOR") combines the aggregate hash of the database file with the hash of the new record. All records are then "added together to provide a highly secure integrity check" for the aggregate database as it changes. A16769(3:17-47); A16763.

Unlike the True Name invention, Fischer is a straightforward example of the "checksum" use long known in the prior art. Although both Fischer and the True Name patents recognize the use of hash functions (including conventional message digest and SHA), Fischer nowhere discloses using content-based identifiers to locate, identify, access, synchronize, replicate, de-duplicate, or move database files (or individual records) within a system containing multiple files and records. To the contrary, and in contrast with the True Name invention, Fischer is simply designed to streamline integrity checks, and its function is consistent with using conventional subjective names for all database management operations.

The PTAB nevertheless held that Fischer (in combination with Woodhill) rendered obvious claims 10 and 21 of the '539 patent. A292.

5. The Satyanarayanan "file system" articles.

The Satyanarayanan articles describe a Unix-based distributed computing environment developed in the mid-1980s at Carnegie Mellon University. By replicating files across multiple servers, the system improved scalability and reliability. But it used conventional pathname and file names, not content-based identifiers, for its file-management operations. A25466-78; A25961-73.

Again, notwithstanding the core differences between these articles and the True Name invention, the PTAB concluded that the Satyanarayanan articles (in combination with the Kantor user manual) rendered obvious claim 30 of the ’662 patent, A392, and claims 1, 2, 81, and 83 of the ’096 patent, A461.

IV. SUMMARY OF THE ARGUMENT

1. *Claim construction*. The PTAB committed three errors of claim construction in these consolidated appeals, undermining its determinations of unpatentability for each of the affected claims.

First, the PTAB misconstrued the structure for the means-plus-function term “identity means” in the ’791 patent. The law holds—indeed, the PTAB’s own rules mandate—that the construction for such a term “must identify the specific portions of the specification that describe the structure … corresponding to each claimed function.” IPR RULE 42.104(3). Both PersonalWeb and EMC identified specific portions of the specification that identified, among other things, five properties that the structure corresponding to the “identity means” function “must have.” A18-19; A2545. These five “must”-have properties guaranteed *at least* cryptographic-hash strength for the “identity means” structure. But rather than adopt these specific portions of the specification, as it was required by law to do, the PTAB composed its own “corresponding” structure: a generic “hash function, *e.g.*, MD5 or SHA.” A20. The PTAB’s adoption of this overly broad structure, found nowhere in the written description of the ’791 patent, was reversible error.

Second, the PTAB misconstrued the function for the means-plus-function term “existence means” in the ’791 patent. The law holds that the function adopted for a means-plus-function claim term cannot be different from the function explicitly recited in the claim. The PTAB initially identified the proper function for this term in its institution decision: “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” A20-22. But when it became clear that EMC had identified no prior art that performed this function, the PTAB rewrote the function to support an unpatentability determination: “the claimed function associated with the ‘existence means’ simply encompasses determining whether a file exists in a registry or table.” A55. The function as so construed (and applied) by the PTAB is plainly different from the function recited in the claim—another reversible error.

Third, the PTAB misconstrued “sequence of non-overlapping parts, each part consisting of a corresponding sequence of bits” in the ’096 patent. While EMC did not initially seek construction of this term, a dispute regarding its scope arose during the IPR, and the PTAB construed it in the final written decision. A469-70. But the PTAB erred in holding that the claimed “sequence” must be read to encompass "intervening gaps" filled with "interstitial parts," themselves comprised of *non-sequence* bits. A469-70. This construction conflicts with the plain language of the claim itself, which—through use of the well-known term of art, "consisting of"—precludes the presence of additional "interstitial parts" that are not members of the claimed sequences. This also was reversible error.

2. *Anticipation*. The PTAB repeatedly misinterpreted and misapplied the law of anticipation in two fundamental ways in these consolidated appeals.

First, the PTAB misapplied the rule of *Net MoneyIN* on multiple occasions, reflecting an entrenched misunderstanding of the law that threatens a far-reaching impact. This rule requires that an anticipatory reference teach "all of the limitations arranged or combined in the same way as recited in the claim." *Net MoneyIn, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008). It is thus impermissible, for anticipation purposes, to combine elements disclosed in "separate protocols" of a prior-art reference. *Id.* at 1371. Despite its awareness of this rule, the PTAB repeatedly rearranged and combined separate protocols from Woodhill and Langer to support its anticipation determinations. A77-79; A199-200; A307. The PTAB justified its approach on the grounds that, *e.g.*, the references did not disclaim combining their separate protocols. A307. But that turns the law on its head. The law does not *permit* rearrangement and combination in the absence of disclaimer; it *precludes* rearrangement and combination in the absence of express teaching.

Second, the PTAB misapplied the rule of *Therasense* on multiple occasions, finding anticipation based on hypothetical embodiments of Woodhill and Kantor, rather than on the actual embodiments disclosed in those references. *Therasense, Inc. v. Becton, Dickinson & Co.*, 593 F.3d 1325, 1332-33 (Fed. Cir. 2010) ("a prior art disclosure of individual claim elements that 'could have been arranged' in a way that is not itself described or depicted [is not anticipatory]"). With respect to the "access" element in certain '791 patent claims, for example, Woodhill expressly teaches that its Binary Objects are accessed through a traditional use of file name and location. A2826; A2839; A2843. The PTAB nevertheless found anticipation based on its speculation that the Binary Object "may" instead be accessed (hypothetically) through use of the Binary Object Identifier. A69. It reached this conclusion notwithstanding testimony that EMC's expert had no idea how Woodhill could access the Binary Object without the traditional use of file name and location. A5817. The PTAB likewise engaged in an improper hypothetical analysis with respect to the "not present" element in the '791 patent claims, A83-85, the "providing" element in the '280 patent claims, A125-29, and the "plurality of parts" element in the '544 patent claim, A216-21.

3. *Obviousness*. The PTAB's obviousness determinations in these consolidated appeals also warrant reversal, for two principal reasons.

First, many of these holdings were premised on (or otherwise undermined by) the PTAB's incorrect claim constructions and other legal mistakes—and should be reversed on that basis. Critically, the PTAB never analyzed obviousness using the correct claim constructions for the terms discussed above.

Second, the PTAB failed to properly apply the *Graham* factors. The controlling *Graham* opinion "was cited but its guidance was not applied, resulting in the application of hindsight and speculation." *Jones v. Hardy*, 727 F.2d 1524, 1529 (Fed. Cir. 1984). The PTAB offered only conclusory and sweeping pronouncements regarding the first two *Graham* factors, and failed to make any express finding regarding the third *Graham* factor—the level of ordinary skill in the art—a critical flaw in its obviousness inquiries. A86-91; A140-43; A222-25; A310-42; A394-404; A463-82. The conclusory nature of the PTAB's analysis, combined with the omission of sufficient factual findings demonstrating the application and satisfaction of the *Graham* factors, renders its obviousness determinations unsupportable across the board.

V. ARGUMENT

A. Standard of Review.

PersonalWeb's claim construction issues are subject to *de novo* review in this Court. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1256 (Fed. Cir. 2007) ("This court reviews claim construction without deference.").[3]

While anticipation is a question of fact reviewed for substantial evidence, PersonalWeb's *Net MoneyIN* and *Therasense* issues go to the PTAB's reading of the law of anticipation, and as such those issues are also subject to *de novo* review. *See In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1363 (Fed. Cir. 2004) ("We review the Board's legal conclusions *de novo* and uphold its factual findings if they are supported by substantial evidence."); *Net MoneyIN*, 545 F.3d at 1371 (interpreting 35 U.S.C. § 102 and confirming legal error "to find anticipation by combining different parts of the separate protocols"); *Therasense*, 593 F.3d at 1332-33 (describing rejected approach as "legally erroneous").

Determination of "obviousness under 35 U.S.C. § 103 is a legal conclusion based on underlying facts." *Translogic*, 504 F.3d at 1256. The PTAB's "ultimate

[3] The PTAB construed the claim terms at issue under the "broadest reasonable interpretation" (BRI) standard. A40-41; A462. Should this Court hold that the standard enunciated in *Phillips*, rather than the BRI, governs claim construction in IPR proceedings, then the *Phillips* standard should be applied in this appeal. *See Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). In any event, the contested constructions are erroneous under either standard.

determination of obviousness" is thus reviewed *de novo*, while its "underlying findings of fact receive review for substantial evidence." *Id.* Those underlying findings of fact are governed, however, by application of the *Graham* factors—and PersonalWeb's issue regarding whether the PTAB properly applied the *Graham* factors is another legal question subject to *de novo* review. *See Custom Accessories, Inc. v. Jeffrey-Allan Industries, Inc.*, 807 F.2d 955, 958-59, 961 (Fed. Cir. 1986) (noting that "[w]e must be convinced from the opinion that the district court actually applied *Graham*," and finding legal error where a "fleeting reference to *Graham* does not convince us that the district court in fact properly analyzed obviousness using the *Graham* analysis"); *Jones*, 727 F.2d at 1529 (finding legal error where *Graham* "was cited but its guidance was not applied, resulting in the application of hindsight and speculation").

B. The PTAB Erred in Claim Construction.

1. The PTAB misconstrued the structure for "identity means" in the '791 patent.

The PTAB misconstrued the means-plus-function term "identity means" in claim 1 of the '791 patent by composing its own structure for the claim term, rather than adopting the corresponding structure described in the specification. A55. In rejecting the structure disclosed and linked in the specification—the very structure initially identified by both parties—the PTAB committed reversible error.

a. The corresponding structure for a means-plus-function term must be drawn from the specification.

The construction of a means-plus-function claim term is governed by statute:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

35 U.S.C. § 112, ¶ 6; *In re Donaldson Co.*, 16 F.3d 1189, 1993 (Fed. Cir. 1994).

The first step in construing a means-plus-function element is to identify the function recited in the claim. *Asyst Techs., Inc. v. Empak, Inc.*, 268 F.3d 1364, 1369 (Fed. Cir. 2001). The second step is to "identify the corresponding structure set forth in the written description that performs the particular function set forth in the claim." *Id.* While structural features "that do not actually perform the recited function" are not read into the claim, *id.* at 1370, a "means-plus-function claim encompasses all structure in the specification corresponding to that element and equivalent structures," *Micro Chem., Inc. v. Great Plains Chem. Co.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999). This means that when "multiple embodiments in the specification correspond to the claimed function, proper application of § 112, ¶ 6 generally reads the claim element to embrace each of those embodiments." *Id.; see also Serrano v. Telular Corp.*, 111 F.3d 1578, 1583 (Fed. Cir. 1997) ("includes … any alternative structures"); *Cardiac Pacemakers, Inc. v. St. Jude Med., Inc.*, 296

F.3d 1106, 1119 (Fed. Cir. 2002) ("corresponding structure must include all structure that actually performs the recited function").

The rules are no different when the PTAB applies the "broadest reasonable interpretation" standard to a means-plus-function term: "the 'broadest reasonable interpretation' that [the PTO] may give means-plus-function language is that statutorily mandated in paragraph six." *Donaldson*, 16 F.3d at 1194-95. "Accordingly, the PTO may not disregard the structure disclosed in the specification corresponding to such language" when construing means-plus-function terms. *Id.* at 1195. In line with this holding, the rules governing these IPR proceedings make clear that any PTAB construction of a means-plus-function term "*must identify the specific portions of the specification* that describe the structure, material, or acts corresponding to each claimed function." IPR RULE 42.104(3) (emphasis added). Any construction of a means-plus-function term that fails to identify the specific portions of the specification corresponding to the claimed function—for each and every of the disclosed embodiments—is thus erroneous under the "broadest reasonable interpretation" standard.

b. The PTAB failed to draw the corresponding structure for the claimed function from the specification.

Critically, the PTAB failed to identify *any* of the specific portions of the specification that describe the structure corresponding to the claimed function.

The claim term at issue is:

> "identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier …."

A2559 (underlining added). Both the parties and the PTAB agreed that the above-quoted language constitutes a means-plus-function term subject to § 112, ¶ 6, and that the function consists of the underlined portion of the quote. A18; *Asyst*, 268 F.3d at 1369. Both parties were also largely in agreement—at least initially—with respect to the corresponding structure in the specification. That is, both parties generally agreed that the corresponding structure includes a processor programmed to perform the "Calculate True Name" mechanism in accordance with the portions of the specification found at columns 12:54-13:19 and column 14:1-39.[4] A18-19.

Notwithstanding this general agreement among the parties regarding the structure corresponding to the "identity means" functionality, the PTAB ignored both proposals and "identif[ied] the corresponding structure … to be a data processor programmed to perform a hash function, *e.g.*, MD5 or SHA." A20. This

[4] The patent defines a "True Name" as the "substantially unique data identifier." A2542(6:6-8). The "Calculate True Name" mechanism thus corresponds to the identity means for "determining … a substantially unique identifier." A2559. While the parties were largely in agreement on this point, EMC identified some additional features that do not actually perform the recited function. For example, EMC's proposal included a reference to column 31:32-50, A18, but that portion of the specification describes the "Verify True File" mechanism, not the "Calculate True Name" mechanism, A2555.

alleged structure incorporates no citation to any specific portion of the specification, and for good reason: it is not drawn from the specification. Indeed, outside of the "References Cited" section, the phrase "hash function" does not appear *ipsissimis verbis* in the written description of the '791 patent. Furthermore, as discussed above, the specific functions identified as examples in the patent are *cryptographic* hashes—a particular kind of hash with very distinctive characteristics. The PTAB's construction, which simply requires "a hash function," is thus plainly incorrect. IPR RULE 42.104(3).

Corresponding structure is found in two key passages of the specification:

1. The principal structure corresponding to the "identity means" functionality is disclosed at columns 12:54-13:19. Rather than speaking in terms of a "hash function" generically, the specification explains that a "True Name"—defined earlier as the substantially unique identifier, A2542(6:6-8)—is "computed using a function, MD," which "*must* have the following properties:"

> 1. The domain of the function MD *is* the set of all data items. The range of the function MD *is* the set of True Names.
> 2. The function MD *must* take a data item of arbitrary length and reduce it to an integer value in the range 0 to N-1, where N is the cardinality of the set of True Names. That is, for an arbitrary length data block B, 0<=MD(b)<=N.
> 3. The results of MD(B) *must* be evenly and randomly distributed over the range of N, in such a way that simple or regular changes to B are virtually guaranteed to produce a different value of MD(B).
> 4. It must be computationally difficult to find a different value B' such that MD(B)=MD(B').
> 5. The function MD(B) *must* be efficiently computed.

A2545-46(12:54-13:9) (emphases added). The patentees left no doubt that these properties were necessary means of performing the function: they not only used the absolute term "must" when introducing the properties, they used this absolute term repeatedly throughout the description of the properties themselves.

The specification then explains that a "family of functions with the above properties are the so-called message digest functions," which "include MD4, MD5, and SHA," A2546(13:10-14)—cryptographic hashes—and that the "presently preferred embodiments" employ "either MD5 or SHA." A2546(13:14-19).

It is this last statement that the PTAB apparently focused on in erroneously adopting its overly broad "corresponding" structure of a generic "hash function, *e.g.*, MD5 or SHA." A20. While it is true that an MD5 or SHA hash function would meet the required five properties of the claimed means, it is not true that any generic "hash function" would do so—and appending "MD5 or SHA" as permissive "*e.g.*" examples to the PTAB's formulation does not limit the adopted structure in the manner required by the specification.[5] Critically, the PTAB's overly broad structure is unmoored from the controlling passage cited above—it lacks any requirement for the five properties that the specification expressly

[5] Indeed, including permissive "*e.g.*" examples in claim constructions is itself a practice discouraged by the PTO. *See* MPEP 2173.05(d) ("If stated in the claims, examples … may lead to confusion over the intended scope of a claim.").

provides “must” be included in the structure for determining the substantially unique identifier. A2545-46(12:61-13:9).

The PTAB eventually recognized that “the specification of the ’791 patent uses the absolute term ‘must’ when describing MD hash functions generally,” but it reasoned that the five “must”-have properties could be ignored because they were discussed “in the context of ‘preferred embodiments.’” A44. This was error three times over: *First*, the specification states that a “True Name is computed using a function, MD, which … must have the following properties.” A2545(12:55-61). For good measure, the word “must” is repeated throughout the description of these mandated properties. *Second*, the “preferred embodiment” language cited by the PTAB only indicates that, out of the universe of hashes *that have all of the necessary properties*, the “presently preferred embodiments” employ “either MD5 or SHA.” A2546(13:14-19). *Third*, “alternative structures” corresponding to the claimed function must be included in any event. *Serrano*, 111 F.3d at 1583. Even if the “must”-have properties were limited to the preferred embodiments (though they are not so limited), they would still need to be identified as corresponding structure. *Micro Chem.*, 194 F.3d at 1258.

2. The specification provides further corresponding structure at column 14:1-39, which discloses a “mechanism for calculating a True Name … with reference to FIGS. 10(a) and 10(b).” A2546(14:1-3). In this embodiment data

items may be either simple or compound, and the mechanism computes a True Name for a compound data item by: breaking the data item into segments; computing the True Name of each segment; creating an indirect block of the segment True Names; and then computing the True Name of that indirect block. A2546(14:4-31). If the indirect block is itself a compound data item, "the mechanism is invoked recursively until only simple data items are being processed." A2546(14:32-35). This recursive mechanism thus performs a particular "hash of hashes" algorithm to determine the substantially unique identifier for a compound data item. A2546; A199.

Although both parties initially identified this portion of the specification as providing additional corresponding structure to the "identity means" functionality, the PTAB declined to include this passage in its construction because "PersonalWeb does not explain adequately why the steps illustrated in that embodiment are necessary to perform the claimed function." A45. In support, the PTAB highlighted the *Micro Chemical* rule that a construction should not incorporate "structure from the written description beyond that necessary to perform the claimed function." A45; *Micro Chem.*, 194 F.3d at 1258. But the PTAB again misunderstood the law—*Micro Chemical* itself makes clear that a legally proper construction "encompasses all structure in the specification corresponding to that element," including "alternative embodiments of the invention." *Id.* Structure that is not "necessary" for these purposes is structure that does not *actually perform* the recited function. *Id.*; *Cardiac Pacemakers*, 296 F.3d at 1119. The mechanism for calculating a True Name for compound data items may not be absolutely necessary to the invention, but when this embodiment is in play, the structure that corresponds to the "identity means" functionality is disclosed at columns 14:1-39. That portion of the specification thus should also have been included in the construction of this term. *Serrano*, 111 F.3d at 1583.

The PTAB's failure to identify the specific portions of the specification corresponding to the claimed function is a plain violation of IPR RULE 42.104(3), as well as the statutory and decisional authorities on which that rule is based. This failure is further highlighted by the PTAB's markedly different treatment of the corresponding structure for the *other* means-plus-function claims at issue: the PTAB construed three more terms subject to § 112, ¶ 6, and for each of these it identified a corresponding structure that incorporated specific references to the written description.[6] The PTAB simply failed to do that for the "identity means" term, requiring a rejection of its construction.

[6] For "existence means," the PTAB identified the corresponding structure as "a data processor programmed according to step S260 illustrated in Figure 14." A23. For "associating means," it identified the corresponding structure as "a data processor programmed according to the steps S230, S232, and S237-S239 illustrated in Figure 11." A25. And for "access means," it identified the corresponding structure as "a processor programmed according to steps S292 and S294 illustrated in Figure 17(a)." A26.

c. The PTAB's error was harmful.

Significantly, the PTAB's failure to limit the "identity means" term to the corresponding structure disclosed in the specification was reversible error. The Binary Object Identifier in Woodhill consists of four appended fields: (1) 32 bits representing the Binary Object size; (2) 32 bits resulting from application of a cyclic redundancy check algorithm; (3) 32 bits resulting from application of a longitudinal redundancy check algorithm; and (4) 32 bits resulting from application of an undisclosed "hash" algorithm. A2841. There is no evidence that this four-part identifier meets all five of the mandatory criteria for a True Name described in columns 12 and 13 of the '791 patent. Certainly these appended fields do not constitute a cryptographic hash. A5934-35. Woodhill has no need for cryptographic strength because the Binary Objects there are relatively small, and are nearly all of fixed size. A2839-41. Further, there is no evidence that Woodhill's four-part identifier reads on the recursive mechanism for determining the True Name of compound data items described in column 14 of the '791 patent.[7]

[7] EMC may argue that the PTAB later held that modifying Woodhill to use "an MD5 hash algorithm" would have been "an obvious improvement." A89. As discussed below, this holding was erroneous and insufficiently supported, but in any event it is of no moment—as the PTAB never suggested that it would have been obvious to somehow modify Woodhill such that it used the recursive algorithm disclosed at column 14:1-39 of the '791 patent, A86-90.

The PTAB's error in construing the means-plus-function "identity means" term thus necessitates reversal of its determinations of unpatentability with respect to claims 1-4 and 29 of the '791 patent. A101.

2. The PTAB misconstrued the function for "existence means" in the '791 patent.

The PTAB improperly composed its own *structure* for the means-plus-function term "identity means," but it improperly composed its own *function* for the means-plus-function term "existence means." A52-55. While the PTAB correctly identified the "existence means" function in its institution decision—"determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items," A22—the PTAB later rewrote this functional claim language such that "the claimed function associated with 'existence means' simply encompasses determining whether a file exists in a registry or table." A55. These functions are not the same, and in rewriting the claimed function, the PTAB again committed reversible error.

a. The function for a means-plus-function term cannot be different from that explicitly recited in the claim.

As noted above, the first step in construing a means-plus-function element is to identify the function recited in the claim. *Asyst Techs.*, 268 F.3d at 1369. The law is further clear that the "statute does not permit limitation of a means-plus-function claim by adopting a function different from that explicitly recited in the claim." *Micro Chem.*, 194 F.3d at 1258; *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314 (Fed. Cir. 2003). And again, the rule is no different when the PTAB applies the "broadest reasonable interpretation" standard to a means-plus-function term. *In re Teles AG Informationstechnologien*, 747 F.3d 1357, 1367-68 (Fed. Cir. 2014); *see also Donaldson*, 16 F.3d at 1194-95.

b. The PTAB erroneously rewrote the function explicitly recited in the claim.

Both the parties and the PTAB agreed that "existence means" constitutes another means-plus-function term subject to § 112, ¶ 6: existence means for "determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items." A20-21. In its institution decision, the PTAB identified both the claimed function and the specification-based corresponding structure for this means-plus-function term:

> We identify the corresponding structure for performing the recited function—namely "determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items"—to be a data processor programmed according to step S232 illustrated in Figure 11 or step S260 illustrated in Figure 14.

A22. PersonalWeb agreed that the PTAB had properly identified the "existence means" function as it was explicitly recited in the claim, and had further properly identified the specific portions of the specification that provided structure corresponding to the claimed functional language. A2274-76.

But after the PTAB issued the constructions in its institution decision, PersonalWeb was allowed to present expert evidence as to the deficiencies of the prior art.[8] This evidence confirmed that EMC had failed to identify any prior art that determined "whether a particular data item is present in the system, by examining the identifiers of the plurality of data items." That should have resulted in a determination of patentability for the challenged claims. It did not. Rather than asking whether the structures identified by EMC satisfied the "existence means" function that had been identified in the institution decision, the PTAB inexplicably revisited its initial construction in its final written decision—and simply rewrote the claimed function to support an unpatentability determination. A51-55. Citing the *Markman* briefing in a related case, the PTAB explained that it was "not persuaded by PersonalWeb's arguments because they are based on an overly narrow claim construction." A53. The PTAB then rewrote the function in dispute.

It started with the first half of that functional phrase:

> Contrary to PersonalWeb's arguments, the claimed function of "determining whether a particular data item is present in the system" does not encompass searching all the files in a system. Instead, according to the specification of the '791 patent, it simply includes determining whether a file exists in a registry or table.

[8] IPR proceedings do not provide an equal opportunity for competing experts to be heard prior to the institution decision. While the petition for IPR may be supported by an expert declaration, the patent owner's preliminary response may not include such testimonial evidence. 37 C.F.R. § 42.107(c).

A53-54. In the next step of its analysis, the PTAB expanded this construction to cover the entire "existence means" function:

> [a]s we explained above, the claimed function associated with the "existence means" simply encompasses determining whether a file exists in a registry or table.

A55. This was plainly erroneous as a matter of law.

The PTAB was required to adopt and apply the "existence means" function as it was "explicitly recited in the claim." *Micro Chem.*, 194 F.3d at 1258. Adopting a function that is "different" in any way contravenes the statute. *Id.* Critically, "determining whether a data item is present *in the system*, *by examining the identifiers of the plurality of data items*," is plainly not the same as simply "determining whether a file exists *in a registry or table*." And the PTAB's citation to the specification of the '791 patent does not save its construction—this Court has held that rewriting a claimed function to account for structural language in the specification is legal error. *Teles AG*, 747 F.3d at 1367. These rules make sense in this case: a specific kind of *registry or table* might be used as *structure* to determine whether a data item is present *in the system* (by examining the identifiers of the plurality of data items), but that is something quite different from simply using a *registry or table* as *structure* to determine whether a data item is present at one particular location in a *registry or table*. The former reflects the innovative approach of the True Name invention; the latter reflects the manner in which the PTAB was forced to rewrite the invention in order to find unpatentability.

The function as recited in the claim requires determining "whether" a data item is "in the system, by examining the identifiers of the plurality of data items." As PersonalWeb demonstrated to the PTAB, the plain and ordinary meaning of "whether" is to "decide or settle conclusively and authoritatively." A2281 (citing the American Heritage Dictionary). And determining "whether" a data item is "in the system" necessarily must be done on a system-wide basis: anything short of a system-wide determination would not conclusively settle "whether" a data item is "in the system." The PTAB's construction of this functional phrase—replacing "in the system" with "in a registry or table," and completely ignoring the additional "by examining the identifiers" functional language—was legal error.

c. The PTAB's error was harmful.

The PTAB's misconstruction of this means-plus-function term was again reversible error. Indeed, the reason that the PTAB had to rewrite the functional phrase was because Woodhill, EMC's principal prior-art reference, has no capability to "determine whether a data item is present in the system, by examining the identifiers of the plurality of data items." As discussed above, Woodhill simply makes a one-to-one comparison of the current Binary Object Identifier to the next most recent Binary Object Identifier for the particular Binary Object at issue. Woodhill provides no system-wide determination of any kind.

EMC's expert, Dr. Clark, conceded this point in his first deposition:

> Q. So when Woodhill is backing up a particular binary object in a given file, Woodhill can only figure out, or tries to figure out, whether that binary object is in that particular file in a previous version, right?
> … A. Yes.
> …
> Q. So Woodhill, when Woodhill is backing up a binary object for a file A, Woodhill has no way of figuring out whether that binary object is in all the other files of the system, right?
> … A. I would say my understanding is that he doesn't have a way of seeing if the data of that binary object matches the data of some binary object not in—that is in some other file.

A5788-90. Dr. Clark reiterated and explained further in his second deposition:

> Q. Can the contents of a binary object exist in multiple files?
> A. Oh, yes.
> Q. So in Woodhill, the contents of a binary object could be in 10 different files?
> A. Yes.
> …
> Q. [I]f there are 100 files at the remote backup server 12 and Woodhill is backing up a binary object for file 20 … Woodhill cannot determine if the contents of that binary object are in any of the files 1 through 19 or 21 through 100 at the remote backup server 12, right?
> … A. The system does not do that.
> Q. So … Woodhill cannot figure out if the contents of a given binary object exists in other files in the system, right?
> … A. That's right.

A6141-43. EMC's expert thus agrees that Woodhill has no capability to determine "whether" a particular data item is present "in the system"—and certainly not "by examining the identifiers of the plurality of data items." A22.

The PTAB concluded that "Woodhill discloses a structural equivalent to the corresponding structure for the claimed 'existence means' because it performs an identical function in substantially the same way to achieve substantially the same results." A57. But that conclusion was based on its flawed construction of the function, which erroneously swapped out "in the system" for "in a registry or table," and ignored entirely the remainder of the function: "by examining the identifiers of the plurality of data items." A51-57. Properly viewed, Woodhill and the True Name invention use different structure, operating differently, to achieve a different result. Woodhill does not perform the function associated with "existence means" at all, because Woodhill has no capability to "determine whether a data item is present in the system, by examining the identifiers of the plurality of data items." As discussed above, Woodhill simply makes a one-to-one comparison of the current Binary Object Identifier to the next most recent Binary Object Identifier for the particular Binary Object at issue. The ways and results of performing these different functions are fundamentally different as well, as discussed above.

The PTAB's misconstruction of this phrase was reversible error, cutting across its findings of anticipation and obviousness. A85-89. The True Name invention teaches system-wide determination, and as EMC's expert made clear, Woodhill "does not do that." A6142. The PTAB's determination of unpatentability with respect to claims 1-4 and 29 of the '791 patent should be reversed.[9]

3. The PTAB misconstrued "sequence of non-overlapping parts" in the '096 patent.

The PTAB also misconstrued the term "sequence of non-overlapping parts, each part consisting of a corresponding sequence of bits," found in claim 1 of the '096 patent, when it held that such a "sequence" must be read to encompass "intervening gaps" filled with *non-sequence* bits. A469-70. The PTAB erred in holding that such intervening *non-sequence* bits—which the PTAB termed "interstitial parts"—could be ignored in the search for an otherwise compliant "sequence of non-overlapping parts." A469-70.

a. The PTAB engaged in an *O2 Micro* construction.

EMC did not initially seek construction of the term "sequence of non-overlapping parts" in its petition for IPR on the '096 patent. A444. But as the IPR proceeded, it became clear that there was a dispute between the parties regarding the scope of that claim term. A24713-15; A24787-88. It was therefore appropriate and necessary for the PTAB to resolve that dispute through construction of the term. *See, e.g.*, *O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351,

[9] The PTAB applied the same erroneous analysis to the similar means-plus-function element ("local existence means") found in dependent claim 2. For the same reasons discussed above, the PTAB's construction of "local existence means" in claim 2 was erroneous and harmful. A62-64.

1362 (Fed. Cir. 2008) ("When the parties present a fundamental dispute regarding the scope of a claim term, it is the court's duty to resolve it."). And that is what the PTAB eventually did. A469.

In its final written decision, the PTAB noted that resolution of a fundamental dispute regarding the obviousness of certain '096 patent claims over the Kantor user manual "depend[ed] on the meaning of 'sequence of non-overlapping parts' whereby a sequence cannot have any intervening gaps." A469. The PTAB resolved the dispute by concluding that such a "sequence" should be read not only to allow "intervening gaps," but to further permit "interstitial parts" filling those gaps. A469. In short, the PTAB construed a "sequence of non-overlapping parts" to allow and encompass intervening "interstitial [non-sequence] parts." A469-70. Because the PTAB expressly resolved "the meaning of" this claim term in its final written decision, A469, that holding is properly subject to review in this Court. *Holmer v. Harari*, 681 F.3d 1351, 1356 n.3 (Fed. Cir. 2012); *see also Lebron v. Nat'l R.R. Passenger Corp.*, 513 U.S. 374, 379 (1995) (noting that review of an issue is proper "so long as it has been passed upon" by the lower tribunal). While the PTAB was right to engage in claim construction for this term, it was wrong in its construction—another reversible error.

b. The PTAB erroneously held that a "sequence of [parts] need only look at the [parts]."

As noted, claim 1 of the '096 patent recites a "data item consisting of a sequence of non-overlapping parts, each part consisting of a corresponding sequence of bits." A25196. The parties disputed whether such a sequence of parts, each consisting of a sequence of bits, could encompass intervening "interstitial parts"—that is, parts (or "non-parts") also consisting of bits but not members of the sequence. The dispute was brought to a head by EMC's argument that the "zip" and "inner" files in Kantor fell within the scope of this claim language. A466-68. That is, EMC proposed that these Kantor structures satisfied the "data item" limitation notwithstanding that they consisted of a collection of "files" (the alleged "parts" included in the sequence) interspersed with a substantial number of "non-files" (the alleged "interstitial parts" or "non-parts" not included in the sequence) comprising, among other things: names, dates, compression ratios, ordering information, zipped paths, comments, headers, and directories.[10] A25224.

[10] EMC was forced to make the distinction between files and non-files because claim 1 further recites applying a first hash function to "*all of the bits* in the sequence," A25196 (emphasis added)—and Kantor hashes only the bits comprising what EMC identified as the "files," and not the bits comprising what the PTAB termed the "non-files." A468. EMC's argument regarding Kantor thus could only stand if the "sequence of non-overlapping parts" in claim 1 was construed to allow and encompass a collection of (bit-based) "parts" interspersed with (bit-based) "non-parts." A469-70. That is, EMC's argument could stand only if the PTAB were free to ignore the fact that Kantor did not teach hashing "*all of the bits*" in the alleged data item. A25196 (emphasis added); A470.

PersonalWeb pointed out that a "sequence of [bit-based] parts" cannot be read to encompass intervening (and also bit-based) non-parts. A24712-27. But in resolving the dispute over the scope of this term, the PTAB disagreed.

According to the PTAB, a "sequence of … parts" must be read to encompass "interstitial parts" (or "non-parts"), such that the alleged "sequence of … files in Kantor can be a sequence, even if they have intervening 'non-files' between them." A469-70. This is a head perplexing. A sequence of "parts" does not encompass "non-parts." The PTAB primarily supported its holding with a "line of people" analogy, reasoning that a "sequence of persons need only look at the persons." A469-70. According to the PTAB, in other words, one determines whether there is a sequence of X by ignoring the intervening presence of anything that is not X. This analysis improperly makes a sequence out of any collection of disparate items: a sequence of persons would encompass intervening monkeys, for example, because a "sequence of persons need only look at the persons." And (taking a "metadata" example) a sequence of novels would encompass intervening dictionaries, because a "sequence of novels need only look at the novels." This is not the plain and ordinary meaning of the term "sequence of non-overlapping parts"—as EMC's expert agreed. A5741-42 (noting that a "sequence" requires "one right after the other"); A5739.[11] And the PTAB pointed to nothing in the intrinsic record that might support its construction of this term. A469-70.

The error in the PTAB's construction is further confirmed by reference to the claim language itself. *See Phillips*, 415 F.3d at 1314 ("the claims themselves provide substantial guidance as to the meaning of particular claim terms"). Claim 1 recites a data item "consisting of" a sequence of parts, each part "consisting of" a sequence of bits. A25196. The phrase "consisting of" is a "term of art in patent law signifying restriction and exclusion"—it means that the patentee "claim[s] what follows and nothing else." *Vehicular Techs. Corp. v. Titan Wheel Int'l Inc.*, 212 F.3d 1377, 1383 (Fed. Cir. 2000). The data item in claim 1 therefore consists *exclusively* of a sequence of parts, and each part consists *exclusively* of a sequence of bits. This claim language precludes the presence of "interstitial parts" that are not members of the claimed sequences. The PTAB's construction, which permits the claimed data item to include myriad bits that are not members of the claimed

[11] The PTAB reasoned that, in determining whether there was a sequence of people, "[o]ne would not need to examine the intervening air, or mosquitoes, or dust that exists between the persons, because those elements would not be people." A470. Whatever the merits of that observation, it has no relation to the issue here, where there is no dispute that both the alleged "parts" and the alleged "interstitial parts" (or "non-parts") are made up of sequences of bits—the fundamental items called out in the claim. The PTAB's holding does not ignore "air," or "mosquitoes," or "dust"; it ignores actual sequences of bits interspersed between other actual sequences of bits. A468.

sequences, thus not only conflicts with common sense; it contravenes the plain language of the claim itself.

In its argument to the PTAB, EMC suggested that PersonalWeb's reading of claim 1 excludes the preferred embodiment as disclosed in column 3 of the '096 patent. A24789; A25179(3:52-59). EMC is wrong. That passage expressly provides that "the identity of the data item depends on all of the data in the data item and only on the data in the data item." A25179(3:54-55). PersonalWeb's construction is consistent with this passage; EMC's construction is not. EMC's construction contravenes this embodiment by making the identity of the data item depend on *some* of the bits in the data item, rather than on *all* of the bits in the data item. A24712-27. PersonalWeb's construction is also consistent with the teaching that "the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data." A25179(3:55-59). Take, for example, EMC's proposal that the "data item" in Kantor is a zip file. PersonalWeb's construction would not make the identity of that zip file dependent upon the name, origin, location, or address *of the zip file*. It would simply make the identity of the zip file dependent upon all of the data in the zip file. This is nothing more than what the patent expressly teaches.

c. The PTAB's error was harmful.

The PTAB's contrary conclusion again constituted reversible error—indeed, the PTAB made clear that its resolution of the unpatentability issue depended upon "the meaning of 'sequence of non-overlapping parts.'" A469. Only by construing the claimed "sequence" to encompass "intervening gaps" filled with *non-sequence* "interstitial parts" could the PTAB find that Kantor rendered unpatentable the challenged claims of the '096 patent. A469-70. Because the PTAB misapprehended the meaning of this term, its determination of unpatentability with respect to the challenged'096 patent claims should be reversed. A509.

C. The PTAB Misapplied the Law of Anticipation.

1. The PTAB repeatedly contravened *Net MoneyIN* by combining separate protocols, not arranged as in the claims, to find anticipation.

The PTAB misinterpreted and misapplied the rule of *Net MoneyIN* on three separate occasions in these consolidated appeals, reflecting an entrenched misunderstanding of the law that threatens to have a substantial and negative impact far beyond these particular proceedings. A77-79; A199-200; A307.

Net MoneyIn holds that,

> unless a reference discloses within the four corners of the document not only all of the limitations claimed but also *all of the limitations arranged or combined in the same way as recited in the claim*, it cannot be said to prove prior invention of the thing claimed and, thus, cannot anticipate under 35 U.S.C. § 102.

Net MoneyIn, Inc. v. VeriSign, Inc., 545 F.3d 1359, 1370 (Fed. Cir. 2008) (emphasis added). Any other rule would improperly "treat[] the claims as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning." *Id.* (citation omitted). Requiring a prior-art reference to disclose every element arranged as in the claims forecloses a finding of anticipation on the inadequate basis of "multiple, distinct teachings that the artisan might somehow combine to achieve the claimed invention." *Id.* at 1371. Instead, this rule of law requires, for anticipation purposes, "a teaching with respect to the entirety of the claimed invention" in a single reference. *Structural Rubber Prods. Co. v. Park Rubber Co.*, 749 F.2d 707, 716 (Fed. Cir. 1984). Thus, if the reference does not "clearly and unequivocally" teach the entire invention "without any need for picking, choosing, and combining various disclosures," then it does not anticipate. *In re Arkley*, 455 F.2d 586, 587-88 (C.C.P.A. 1972).[12]

Under the rule of *Net MoneyIN*, the "disclosure of individual claim elements that 'could have been arranged' in" the claimed manner is insufficient if the claimed arrangement "is not itself described or depicted" in the reference. *Therasense, Inc. v. Becton, Dickinson & Co.*, 593 F.3d 1325, 1332 (Fed. Cir.

[12] While EMC had the burden to prove anticipation "by a preponderance of the evidence" in these IPR proceedings, 35 U.S.C. § 316(e), *Arkley*—a case addressing proceedings in the PTO—holds that this burden cannot be carried absent proof that a single prior-art reference "clearly and unequivocally" teaches the entire invention without any need for combining various protocols. 455 F.2d at 587-88.

2010); *see also Arkley*, 455 F.2d at 588-89 ("We do not read into references things that are not there."). In *Net MoneyIN* itself, the Court held that an "Internet payment system" was not anticipated by a prior-art reference that disclosed all of the elements of the claimed invention—but in "two separate protocols," neither of which "contain[ed] all five [elements] arranged or combined in the same way as claimed in the … patent." 545 F.3d at 1371; *see also Wm. Wrigley Jr. Co. v. Cadbury Adams USA LLC*, 683 F.3d 1356, 1361 (Fed. Cir. 2012). The Court confirmed that it was improper, for anticipation purposes, to combine the elements disclosed in these separate protocols. *Net MoneyIn*, 545 F.3d at 1371.

In these IPRs the PTAB looked to references—Woodhill and Langer in particular—that similarly disclosed multiple distinct protocols, none of which contained all of the elements arranged in the same way as in the challenged claims of the '791, '544, and '539 patents. While ostensibly acknowledging the rule of *Net MoneyIN*, the PTAB held that the distinct protocols could be combined for anticipation purposes because: with respect to Woodhill, the separate protocols were part of an overarching computer program; and with respect to Langer, the reference did not disclaim combining the protocols. A77-79; A199-200; A307. The PTAB erred each time, and those errors require reversal with respect to the claims at issue—claims 30-32 and 41 of the '791 patent; claim 1 of the '544 patent; and claims 10 and 21 of the '539 patent.

a. The PTAB erroneously combined Woodhill's audit and backup protocols.

Claim 41 of the '791 patent depends from claim 30, which recites an underlying method step of "accessing a data item in the system *using the identifier* of the data item." A2560 (emphasis added). Claim 41 recites, in turn:

> 41. The method of claim 30, *wherein said accessing further comprises*: for a given data identifier and for a given current location and a remote location in the system:
>
> determining whether the data item corresponding to the given data identifier is present at the current location, and
>
> based on said determining, if said data item is not present at the current location, fetching the data item from a remote location in the system to the current system.

A2562 (emphasis added). As the elements are arranged, therefore, claim 41 requires "accessing … using the identifier," where that same accessing step further comprises "determining … and … fetching" as recited.

EMC offered Woodhill as anticipating art for this claim, but it had no argument that any of Woodhill's distinct protocols individually satisfied all of the "accessing," "determining," and "fetching" elements required for claim 41. So EMC urged the PTAB to combine Woodhill's separate "audit" and "backup" protocols for anticipation purposes. A77-78. EMC argued that the PTAB could find that Woodhill's audit procedure satisfied the underlying "accessing" element, and that Woodhill's separate backup procedure satisfied the dependent "determining" and "fetching" elements. A77-79. And that is what the PTAB did.

With respect to the "accessing" element, the PTAB agreed

> with EMC that Woodhill's *self-auditing procedure*, which includes using Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25, *describes the function of* "*accessing* a particular data item using the identifier of the data item" ….

A69 (emphases added); A75.[13] And with respect to the "determining" and "fetching" elements, the PTAB again agreed with EMC that these could be satisfied by "Woodhill's *backup procedure*," during which "the data processing system only backs up changed binary objects since the previous backup." A76 (emphasis added). The PTAB thus rearranged and combined the protocols to find anticipation by Woodhill. A77-78.

The PTAB was aware that this approach conflicted with *Net MoneyIN*. Citing the case, the PTAB acknowledged that:

> PersonalWeb contends that EMC relies upon the self-auditing procedure disclosed in Woodhill to describe the "accessing" method step recited in independent claim 30, yet EMC relies upon the backup procedure disclosed in Woodhill to describe the additional features of the same 'accessing' method step recited in dependent claim 41.

A77. But the PTAB nevertheless held that it was permissible to combine the separate protocols in Woodhill on the ground that:

[13] As discussed both above and below, the PTAB erred in concluding that Woodhill uses the Binary Object Identifier to access a Binary Object in its audit procedure. But even if that conclusion had been right, the PTAB erred in rearranging and combining Woodhill's audit and backup procedures in order to find anticipation of the relevant claims. *Net MoneyIn*, 545 F.3d at 1371.

> Woodhill's backup procedure and self-auditing procedure are not mutually exclusive embodiments, but rather are distinct functions that operate with other functions to form one unitary computer program—namely Woodhill's Distributed Storage Manager program 24.

A78; A2839-40.

Of course, the question under *Net MoneyIN* is not whether the separate protocols are "mutually exclusive," nor whether one master computer program "handles the operations" of both of the separate protocols. A78. The question is whether all of the elements are "arranged or combined in the same way as recited in the claim," *Net MoneyIN*, 545 F.3d at 1370, without "any need for picking, choosing, and combining" separate procedures, *Arkley*, 455 F.2d at 587-88. That is not the case with respect to Woodhill's audit and backup procedures, and the PTAB's approach improperly treated claim 41 and Woodhill as "mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in … claims" 30 and 41 of the '791 patent. *Therasense*, 593 F.3d at 1332.

Indeed, the PTAB itself explained that Woodhill's "Distributed Storage Manager program 24 performs *auditing* and reporting *functions* on a periodic basis in order to ensure that the binary objects, *which already have been backed up*, may be restored." A67 (emphases added). The auditing and backup protocols are thus not only separate in purpose and function; they are also separate in time—and the auditing procedure only runs when the backup procedure is finished. A67. There is no question, therefore, that any "accessing" accomplished by the later-run auditing procedure cannot be imputed to the earlier-run backup procedure. There is further no question, therefore, that the required elements—as the PTAB found them—are not arranged in Woodhill as they are in claim 41. *Cf. E-Pass Techs., Inc. v. 3Com Corp.*, 473 F.3d 1213, 1222 (Fed. Cir. 2007) (noting that for infringement, when "the steps of [a] method claim refer to the completed results of the prior step," the steps must be performed "in [sequential] order").

Put another way, even if the PTAB were correct about the nature and function of the audit and backup protocols (though it was not), Woodhill does not disclose an audit procedure (for accessing) wherein the audit procedure further comprises the backup procedure (for determining and fetching).

The PTAB's conclusion that it could rearrange and combine these separate protocols in Woodhill was error under *Net MoneyIN*, and necessitates reversal of its anticipation determination for claims 30-32 and 41 of the '791 patent.[14]

b. The PTAB erroneously combined Woodhill's backup and granularization protocols.

The PTAB committed the same error when it rearranged and combined Woodhill's "granularization" and "backup" protocols to find anticipation with

[14] Claims 30, 31, and 32, like claim 41, contain "determining" and "accessing" steps—which the PTAB found anticipated by combining the audit and backup protocols as discussed above. A75-78; A2560-61. Thus, for the same reasons that the PTAB legally erred with respect to claim 41, it also legally erred with respect to claims 30-32. Notably, the PTAB did not find these claims obvious. A101.

respect to claim 1 of the ’544 patent. A199-201; A207. That claim recites another version of the specific kind of “hash of hashes” discussed above:

> for a first data item comprising a plurality of parts, … applying a first [hash] function to each part of said first plurality of parts to obtain a corresponding part value for each part of said first plurality of parts, … and … obtaining a first value for the first data item, said first value obtained by applying a second [hash] function to the part values of said first plurality of parts of said first data item.

A11157; A199. As the elements are arranged, therefore, this claim requires (among other things) applying a first hash function to obtain part values, and then applying a second hash function to those part values to obtain a first value—in short, as the PTAB summarized, it requires a “hash of hashes.” A199.

EMC offered Woodhill as anticipating prior art for this claim, but again it had no argument that any of Woodhill’s distinct protocols expressly taught a hash of hashes. So EMC urged the PTAB to combine Woodhill’s separate backup and granularization protocols for anticipation purposes. A199-200. The PTAB again accepted this invitation, committing another error under *Net MoneyIN*. A200.

Unlike the backup and audit protocols discussed above, Woodhill’s granularization protocol does not involve Binary Object Identifiers at all—it thus has nothing to do with the element that EMC identifies as the “substantially unique identifier” in Woodhill. A2844-45. The granularization protocol provides a technique for subdividing large database files into “granules” and then tracking changes to the files at the “granule” level. A2844. The tracking is done through a "shadow file" which contains a "contents identifier" for each "granule" in the Binary Object. A2845(15:22-24). The contents identifier includes "a 32-bit hash number" calculated against the contents of the "granule." A2845(15:24-27). Woodhill's granularization protocol thus describes a first hash function, but nowhere does it teach applying a second hash function to the results of that first function. Nowhere does it teach, that is, a hash of hashes.

Implicitly acknowledging this failure to disclose, EMC argued that Woodhill might meet the hash-of-hashes requirement if the shadow file described in the granularization process were put through the local backup protocol described as a separate and distinct process in Woodhill. A10754-55; A200. According to EMC, combining these two protocols might effectively result in the application of a second hash function (as part of the creation of a Binary Object Identifier) to the results of a first hash function (as part of the creation of a contents identifier). Of course, Woodhill nowhere discusses such a hash of hashes. Yet EMC supported its proposal to combine these separate protocols by citing to Woodhill's note that, with respect to the local backup procedure, "the default operation is to back up all files on all disk drives 19 on the local computer 20." A2840(5:62-63).

The PTAB agreed that Woodhill's backup and granularization protocols could be combined on the basis of this note, and further reasoned, once again, that "Woodhill specifically states that each of the functions performed by the DSM program operates in cooperation with the other functions to form *a unitary computer program*.… The disclosure of Woodhill merely divides the DSM program into several distinct functions for explanation purposes." A200.

Of course, the rule of *Net MoneyIN* is focused precisely on the manner in which a prior-art reference arranges its separate protocols "for explanation purposes." A200; *see Net MoneyIN*, 545 F.3d at 1371. In the absence of "a teaching with respect to the entirety of the claimed invention," there is no anticipation. *Structural Rubber*, 749 F.2d at 716. Furthermore, such teaching must be "clear[] and unequivocal[]." *Arkley*, 455 F.2d at 587-88. There is no teaching, clear or otherwise, of a hash-of-hashes in Woodhill. And the note relied upon by the PTAB, to the effect that "the default operation is to back up all files on all disk drives 19 on the local computer 20," A2840(5:62-63), does not justify its contravention of *Net MoneyIN*. Indeed, while Woodhill discusses this "default operation" with respect to backing up files "on the local computer 20," it makes clear that this operation is *not used* with the granularization protocol:

> This technique of subdividing files into "granules" is only used to reduce the amount of data that must be transmitted to the remote backup file server 12 and *is not utilized in making backup copies of binary objects for storage on local computers 20*.

A2845(15:4-8) (emphasis added).

The PTAB's decision to combine the separate backup and granularization protocols in Woodhill was thus legal error under *Net MoneyIN*—and the magnitude of that error is exacerbated by the fact that Woodhill itself states that the protocols are not to be combined. This error requires reversal of the PTAB's anticipation determination for claim 1 of the '544 patent.[15] A207.

c. The PTAB erroneously combined Langer's package and standalone protocols.

The PTAB committed the same error a third time when it combined the so-called "standalone" and "package" embodiments discussed in Langer to find anticipation with respect to claims 10 and 21 of the '539 patent. A305-07. Claim 10, which is representative, recites in relevant part:

> in response to a request, said request comprising a first identifier, obtaining a plurality of segment identifiers, … using at least one of said segment identifiers …, requesting at least one particular segment of said plurality of segments that comprise the data item … and … obtaining said particular segment ….

A15848; A305-06. As the elements are arranged, the claim requires using a first identifier to request and obtain a plurality of segment identifiers, then using one of those segment identifiers to request and obtain a particular segment of a data item.

This time EMC offered Langer as an anticipating reference. As noted, Langer describes four ideas for using "unique identifiers" to improve FTP systems,

[15] After determining that it could permissibly combine these separate protocols for anticipation purposes, the PTAB noted that it "credit[ed] the testimony of Dr. Clark over that of Dr. Dewar." A202. But of course no expert testimony can justify combining two separate protocols for these purposes, neither of which unequivocally teaches a hash of hashes. That is wrong as a legal matter.

each markedly different from the uses disclosed and claimed in the True Name patents. These four "standalone" uses are addressed under the heading "unique identifiers." A2572; A305-07. Langer addresses "packages" under a separate heading, and there discusses a "related problem" that occurs when "essentially the same collection of information may be available as different [zip] files etc." A2574. Langer does not propose one of its four "unique identifier" solutions for this separate problem—"[u]ltimately these [packages] do have to be regarded as DIFFERENT files and any connections between them listed separately." A2574.

Nevertheless, EMC cobbled together an anticipation argument by combining the distinct standalone and packages discussions in Langer. In particular, EMC argued that Langer discloses, in its standalone discussion, "that a user may submit a query to a database using an MD5 code to determine the location of a file so it could be retrieved." A307. EMC further argued that Langer discloses, in its packages discussion, "that the MD5 code of a package may be used to obtain the concatenated block of MD5 codes, or a listing of MD5 codes and filenames of the inner files of the package." A307. EMC finally argued that, if these two teachings were to be combined—though they were not in Langer—"an MD5 code for a particular inner file of the package may then be used to identify and retrieve that particular inner file," thus meeting the elements of the claims at issue. A307. The PTAB again "agree[d] with EMC." A307.

In response to PersonalWeb's argument that EMC was again improperly urging the PTAB to combine distinct embodiments for anticipation purposes, the PTAB reasoned that "[n]othing in Langer indicates that the unique identifiers … are limited to standalone files, and could not apply to files within a package." A307. In other words, the PTAB held that it was proper to rearrange and combine the various teachings because Langer did not expressly disclaim the rearrangement and combination that was being urged. That approach turns the law on its head. The law does not *permit* rearrangement and combination in the absence of disclaimer; it *precludes* rearrangement and combination in the absence of express teaching. *Net MoneyIN*, 545 F.3d at 1371; *Structural Rubber*, 749 F.2d at 716; *Arkley*, 455 F.2d at 587-88. As the Court held in *Therasense*, "a prior art disclosure of individual claim elements that 'could have been arranged' in a way that is not itself described" is not anticipatory. 593 F.3d at 1332; *see also Arkley*, 455 F.2d at 589 ("We do not read into references things that are not there.").

Langer does not unequivocally teach anything analogous to using an identifier to request and obtain a segment of a data item. Langer thus does not teach "all of the limitations arranged or combined in the same way as recited in the claim," and cannot anticipate under *Net MoneyIN*, 545 F.3d at 1370. The PTAB's contrary conclusion again constitutes legal error and requires reversal of its anticipation determination for claims 10 and 21 of the '539 patent.

2. The PTAB repeatedly contravened *Arkley* and *Therasense* by crediting hypothetical embodiments to find anticipation.

The PTAB further misapplied the law of anticipation, contravening *Arkley* and *Therasense*, when it credited hypothetical elements in the prior art as the basis for its finding of anticipation with respect to claims 4, 29-33, and 41 of the ’791 patent, claims 36 and 38 of the ’280 patent, and claim 1 of the ’544 patent.

a. Contrary to the PTAB’s hypothesizing, Woodhill does not access (or provide) Binary Objects using the Binary Object Identifier.

Claims 4, 29-32, and 41 of the ’791 patent all require a version of “accessing a data item in the system using the identifier of the data item.” A2559-62. Claims 36 and 38 of the ’280 patent require a version of “providing” a “data file” to a client from a request using a hash value of the requested file. A6989. EMC offered Woodhill as the anticipating reference, and pointed to Woodhill’s Binary Object Identifier 74 as the “identifier” disclosed in the ’791 patent, and Woodhill’s Binary Object as the respective “data item” or “data file.” A67-69; A125-29. Woodhill, however, does not teach accessing (or providing) the Binary Object by using the Binary Object Identifier 74. Instead, Woodhill teaches using File Location 38 and File Name 40 in File Identification Record 34 to access a particular file, and then further using Binary Object Stream Type 62 and Binary Object Offset 72 in Binary Object Identification Record 58 to access a particular Binary Object in that file.

Binary Object Identifier 74 plays no role in this disclosed process. A2826; A2839(3:55-63); A2840(5:15-45); A2843(11:67-12:5); A2846(18:11-19).

The PTAB recognized that Woodhill taught accessing (or providing) a "file containing a particular binary object" using "File Location 38 and File Name 40," but the PTAB was "nonetheless … persuaded that EMC has presented sufficient evidence to support a finding that a particular binary object or file also *may* be accessed using its Binary Object Identifier 74." A69 (emphasis added). The PTAB cited to EMC's expert for support:

> Dr. Clark confirms such an operation was routine *because it was old and well-known* to access objects using their identifiers.… We credit Dr. Clark's testimony because it is consistent with a general understanding of how one with ordinary skill in the art *would use* an identifier for basic file management functions, *e.g.*, using an identifier to access a record stored in a database.

A68 (emphases added).[16] The PTAB thus engaged in speculation and simply assumed the presence of the element that it could not find in Woodhill.

Furthermore, while the PTAB relied upon Dr. Clark's affidavit to ignore Woodhill's teaching regarding access using the link to File Location 38, File Name 40, and File Identification Record 34, on cross-examination Dr. Clark admitted that

[16] Significantly, as discussed further below, the final written decision for the '791 patent illustrates that the PTAB never made an express determination of the level of ordinary skill in the art. The PTAB noted PersonalWeb's assertion regarding the level of ordinary skill for the '791 patent, but the PTAB never indicated that it was adopting that standard, or any other standard. A46.

he had no idea how Woodhill might access (or provide) a Binary Object without those expressly disclosed structures:

> Q. Okay … how about if you explain then, how the binary object is accessed in Woodhill's auditing procedure without using the filename, without using the file location, without using the length of File Identification Record? How is that possible?
> … A. I'm not able to do that right now.

A5816.

> Q. Do you know how the binary object is accessed in Woodhill's auditing procedure without using the Link to File Identification Record [34], Filename 40, File Location 38?
> … A. I do not know.

A5817. When pressed, in fact, Dr. Clark admitted that he *did not know* how the Binary Object is accessed in Woodhill's audit protocol:

> Q. Do you know how the binary object is accessed in Woodhill's auditing procedure? (Witness reviewing)
> A. I do not.

A5817.

The PTAB's speculation as to how Woodhill "may" have worked and what one of ordinary skill "would use" was improper, and that impropriety is highlighted by the PTAB's reliance on an expert who admitted under cross-examination that he did not know how the Binary Object is accessed in Woodhill.

For anticipation, a prior-art reference must disclose each limitation in the claim at issue expressly or inherently. Here, Woodhill teaches accessing (or providing) in a way that is very different from the True Name invention, as set forth above, and it was error for the PTAB to engage in speculation as to how Woodhill "may" have been constructed and "would" have operated otherwise.[17] *Therasense*, 593 F.3d at 1332 ("a prior art disclosure of individual claim elements that 'could have been arranged' in a way that is not itself described or depicted [is not anticipatory]"); *Arkley*, 455 F.2d at 589 ("We do not read into references things that are not there."). The PTAB failed to show that these limitations were expressly disclosed, and failed to follow controlling law regarding inherent disclosure in a prior-art reference. *Transclean Corp. v. Bridgewood Servs., Inc.*, 290 F.3d 1364, 1373 (Fed. Cir. 2002) ("anticipation by inherent disclosure is appropriate only when the reference … must *necessarily* include the unstated limitation"); *Therasense*, 593 F.3d at 1332-33. In short, this legal error led the PTAB to improperly credit missing and hypothetical elements in the Woodhill reference.

[17] Despite Dr. Clark's admission that he did not know how Woodhill accesses a Binary Object to restore it to a local computer, A5817, and despite Woodhill's explanation that Binary Objects are accessed using the file name, file location, Binary Object Stream Type, and Binary Object Offset fields (all stored in the File Database for each file), A2843(11:67-12:28), the PTAB rejected claims 36 and 38 of the '280 patent because PersonalWeb failed to prove a negative: "We are not persuaded by PersonalWeb's argument that Woodhill *does not use* Binary Object Identifier 74 … to identify and request a particular binary object." A127 (emphasis added). This was error. Anticipation requires proof that Woodhill *does use* Binary Object Identifier 74 to identify and request a particular Binary Object.

The PTAB's contravention of *Arkley* and *Therasense* necessitates reversal of its anticipation determination for claims 4, 29-32, and 41 of the '791 patent, as well as claims 36 and 38 of the '280 patent. A70-71; A76-79; A101; A125-29; A153.

b. Contrary to the PTAB's hypothesizing, Woodhill does not confirm the absence of a Binary Object on the remote backup server.

Claims 33 and 41 require determining whether a "data item" is "not present" at a "destination" or "location" and (if not present) moving the data item to that location. A2561-62. In comparing Woodhill to these claims, the PTAB identified the "remote backup server 12" as corresponding to the "destination" (claim 33) and "location" (claim 41). A82-85. For both claims, the PTAB "credit[ed] Dr. Clark's testimony," finding that "during Woodhill's backup procedure, Distributed Storage Manager program 24 determines whether a Binary Object or file corresponding to Binary Object Identifier 74 already exists on remote backup server 12 and, if not, transmits the Binary Object or file to that location." A83; A85.

The PTAB erred in reading into Woodhill a capability that it does not disclose. *Arkley*, 455 F.2d at 589. During backup, Woodhill only determines whether the individual Binary Objects of a file that has been modified on a local computer already exist in the file's counterpart stored on the remote backup server:

> Those binary objects that have changed are identified by comparing the Binary Object Identifiers 74 … with the corresponding Binary Object Identifiers 74 associated with the next most recent Backup Instance Record 42 for the file identified by the Backup Queue

> Record 75 currently being processed. The Binary Object Identifiers 74 … are compared against their counterparts in the File Database 25 (e.g., the Binary Object Identifier 74 … that identifies the first binary object in the file (as determined by the Binary Object Stream Type field 62 and the Binary Object Offset field 72) ….

A2842(9:9-22). Thus, Woodhill only compares a Binary Object's Identifier against a single Identifier for the counterpart Binary Object stored on the remote backup server (located using the Stream Type and Offset fields); Woodhill never confirms whether the Binary Object is "not present" elsewhere on the remote backup server. In other words, if the Binary Object Identifier for the modified file does not match the Identifier for its backed-up counterpart, the modified Binary Object replaces the counterpart in backup storage, regardless of whether the same Binary Object exists in the same file or a different file elsewhere on the backup server.[18]

Contrary to the PTAB's finding, Dr. Clark did not state that Woodhill determines whether a Binary Object is "not present" somewhere on the remote backup server. Instead, Dr. Clark only stated that the Woodhill backup procedure

[18] In his declaration, Dr. Clark adjusts the location of quotation marks to make it appear, incorrectly, that Woodhill compares a single, newly-calculated Binary Object Identifier against multiple Binary Object Identifiers. *Clark declaration*: "Woodhill's backup procedure compares a newly calculated Binary Object *Identifier 74* 'with the corresponding Binary Object *Identifiers 74* associated with the next most recent Backup Instance Record 42 …' (Woodhill at 9:11-14; Ex. 1005)." A5136-37 (emphasis added). *Actual Woodhill quote*: "Those binary objects that have changed are identified by comparing the Binary Object *Identifiers 74* … with the corresponding Binary Object *Identifiers 74* associated with the next most recent Backup Instance Record 42 …." A2842 (emphasis added).

determines whether a Binary Object "is present at a *particular location* in the system"—namely, for a modified file, whether it is in the corresponding file on the backup server at the same offset. A2923-24 (emphasis added). Despite Dr. Clark's careful explanation that Woodhill determines only whether a Binary Object is present or not at a "particular location" within the system, the PTAB's final written decision substituted "particular location" (*i.e.*, an offset within a known file in backup) for the more general "destination" or "location" (*i.e.*, the entire backup server itself). A82-83; A85. No evidence supported the PTAB's inexplicable substitution of "particular location" in a single backed-up file with the entire backup file system. Indeed, Dr. Clark readily admitted that Woodhill makes no determination of whether a Binary Object is present or not present outside the particular corresponding offset location within the previously backed-up file:

> Q. So Woodhill cannot figure out if the contents of a given binary object exists in other files in the system or in other binary objects of the same file; is that right?
> … A. So I would say—I would not say he can't do something. I would say his system doesn't do that.

A6144.

Again, the PTAB's crediting of hypothetical elements in Woodhill contravened *Arkley* and *Therasense*, and necessitates reversal of its anticipation determination for claims 33 and 41 of the '791 patent. A81-84; A101.

c. Contrary to the PTAB's hypothesizing, Kantor does not separate a zip file and hash its parts.

The PTAB committed the same error yet again in holding that Kantor anticipated claim 1 of the '544 patent. That claim requires: (a) separating a "data item" into a "plurality of parts"; (b) applying a hash function to each part to obtain a "value" for each part; and (c) applying a second hash function to the "value" of each part to obtain a value for the "data item" as a whole. A11157.

Kantor creates a "contents signature" for "zip files," (a "zcs") but uses a different method. Kantor does not separate a zip file into a plurality of parts and then apply a hash function to each part. Rather, Kantor calculates and records a CRC-32 and file length values for each file before any file is compressed and placed in a zip file (along with overhead information). A2604. After the files are zipped, Kantor takes the pre-zipped CRC-32 value for each file, adds them using a modulo function, and does the same with the pre-zipped file length. The combination of the two resulting numbers forms the "zcs." A2605; A2651.

In finding claim 1 anticipated, the PTAB reasoned that the Kantor zip file corresponds to claim 1's "data item." But the PTAB ignored the fact that Kantor never separates a zip file into constituent parts and applies a hash to each part. Indeed, the PTAB inexplicably found it "irrelevant" that the "zsc" for the zip file is created from the values obtained by applying a CRC-32 to each constituent file *before* any file is compressed, zipped, and the zip-file overhead added—even though the claim language requires the opposite. A216-21. In effect, the PTAB rewrote claim 1 to recite hashing a plurality of files, combining the files with any overhead, and adding together the pre-combination hash results. The PTAB justified this rewrite by finding that "nothing in Kantor limits" a zip file to compressed versions of the pre-zipped files. A218. But the PTAB's reasoning as to how the Kantor software *could* work again misses the point: *Therasense* restricts anticipation to actual or inherent disclosures—and nothing in Kantor actually or inherently discloses the creation of a value for a zip file by separating the zip file into parts and then applying a hash function to each part. Thus, the PTAB's rejection of claim 1 based on Kantor should be reversed. A242.[19]

D. The PTAB Erred in Holding the Challenged Claims Obvious.

1. The PTAB's obviousness determinations were premised on incorrect claim constructions and other legal errors.

Many of the PTAB's determinations of obviousness were premised on (or otherwise undermined by) the incorrect claim constructions and other legal errors described above, and should be reversed on that basis. *See Hologic, Inc. v. SenoRx,*

[19] With respect to each of the anticipation-related issues pressed above, PersonalWeb submits that the PTAB's combining and rearranging of separate prior-art protocols, as well as its crediting of hypothetical prior-art elements, constitute legal errors subject to *de novo* review. But should the Court disagree, PersonalWeb further submits that each of these errors reveals a fundamental lack of substantial evidence supporting the PTAB's anticipation determinations, and reversal is warranted under a substantial-evidence review as well.

Inc., 639 F.3d 1329, 1330, 1339 (Fed. Cir. 2011) (reversing invalidity findings based on erroneous claim construction); *Custom Accessories*, 807 F.3d at 961-63 (reversing invalidity findings based on legally insufficient analysis).

In particular, the PTAB's obviousness determinations with respect to claims 1-4 and 29 of the '791 patent were based on its misconstructions of both the means-plus-function "identity means" and "existence means" terms. Aside from a brief discussion of secondary considerations and whether a person of ordinary skill would have combined Woodhill with an MD5 hash function, the PTAB's obviousness analysis for claims 1-4 and 29 of the '791 patent consists solely of the observation that there "are no deficiencies in Woodhill to cure." A86-91. The PTAB never analyzed obviousness, in other words, using the proper claim constructions. Similarly, the PTAB never analyzed obviousness for claims 1, 2, 81, and 83 of the '096 patent using the proper plain-and-ordinary-meaning construction of the term "sequence of non-overlapping parts." A469-70. The rejection of these claims thus cannot stand. *Hologic*, 639 F.3d at 1330.

The PTAB's *Net MoneyIN* and *Therasense* errors, while ostensibly directed to anticipation, also infected and undermined its obviousness determinations for the relevant claims. For example, in its anticipation analysis, the PTAB contravened *Therasense* by crediting hypothetical "access" and "not present" elements in Woodhill—and never addressed those missing elements in its obviousness analysis, simply positing that there "are no deficiencies in Woodhill to cure." A86-91. The PTAB's *Therasense* error thus further undermined its obviousness determinations for claims 4 and 29 of the '791 patent. (EMC never asserted, and the PTAB never found, that claims 30-32 and 41 of the '791 patent were obvious.) Similarly, in its anticipation analysis, the PTAB contravened *Net MoneyIN* by combining, *e.g.*, the backup and granularization protocols of Woodhill to find a teaching regarding the required "hash of hashes." A199-200. But the PTAB never addressed that deficiency in its obviousness analysis with respect to claim 1 of the '544 patent—it simply asked whether Woodhill and Kantor might be combined to satisfy a "plurality of parts" limitation. A222-23. This *Net MoneyIN* error thus also undermined the PTAB's obviousness holding for claim 1 of the '544 patent. The PTAB's determinations of unpatentability for these challenged claims should be reversed on this basis. *Custom Accessories*, 807 F.3d at 961-63; *Gechter v. Davidson*, 116 F.3d 1454, 1460 (Fed. Cir. 1997).

These errors necessitate, at the very least, reversal and remand. But PersonalWeb submits that, because the PTAB cannot reasonably be expected to find these missing elements in the prior art on remand, judgment of patentability on these challenged claims would also be appropriate. *See, e.g.*, *Finisar Corp. v. DirecTV Group, Inc.*, 523 F.3d 1323, 1333 (Fed. Cir. 2008) (entering judgment);

CardSoft (Assignment for the Benefit of Creditors), LLC v. VeriFone, Inc., No. 14-1135, 2014 U.S. App. LEXIS 19976, at *12 (Fed. Cir. Oct. 17, 2014) (same).

2. The PTAB failed to properly apply the *Graham* factors.

While the PTAB's particular legal errors undermine its obviousness determinations on many of the challenged claims, its failure to properly apply the *Graham* factors undermines its obviousness determinations across the board.

Obviousness is a question of law that is based on four mandatory factual inquiries, known as the *Graham* factors:

> (a) the scope and content of the prior art; (b) the differences between the prior art and the claims at issue; (c) the level of ordinary skill in the art; and (d) objective evidence of nonobviousness.

Custom Accessories, 807 F.2d at 958; *Graham v. John Deere Co. of Kan. City*, 383 U.S. 1, 17-18 (1966). "Under *Graham*," a lower tribunal must make "proper fact findings on those four inquiries and then assess the ultimate legal question of nonobviousness." *Custom Accessories*, 807 F.2d at 958.

While a tribunal has power to weigh the facts, it has no power to depart from the mandated factors—and it is further obligated to communicate its findings on these factors with sufficient clarity to permit meaningful review by this Court:

> we must be convinced from the opinion that the [lower tribunal] actually applied *Graham* and must be presented with enough express and necessarily implied findings to know the basis of the [lower tribunal's] opinion.

Loctite Corp. v. Ultraseal, Ltd., 781 F.2d 861, 873 (Fed. Cir. 1985); *Custom Accessories*, 807 F.2d at 961. As the Court has explained, the "ultimate test of the adequacy of findings is whether they are sufficiently comprehensive and pertinent to the issue to form a basis for the decision (and whether they are supported by the evidence)." *Loctite*, 781 F.3d at 873. The need for comprehensive and "express *Graham* findings takes on an especially significant role" in patent appeals because of an "occasional tendency" for lower tribunals "to depart from the *Graham* test, and from the statutory standard of unobviousness that it helps determine, to the tempting but forbidden zone of hindsight." *Id.*

The PTAB's obviousness analyses in these IPRs fell short of this required standard. As in *Loctite* and *Jones*, the *Graham* opinion "was cited but its guidance was not applied, resulting in the application of hindsight and speculation." *Jones*, 727 F.2d at 1529 ; *Loctite*, 781 F.2d at 873. And as in *Custom Accessories*, the PTAB's "fleeting reference to *Graham*" should not convince the Court that the PTAB "in fact properly analyzed obviousness using the *Graham* analysis." 807 F.2d at 958; A86-91; A140-43; A222-25; A310-42; A394-404; A463-82.

The PTAB's treatment of obviousness for the lead '791 patent is representative. The PTAB started with a citation to *Graham*, noting that it would "analyze the ground of unpatentability based on obviousness over Woodhill with the above-identified principles in mind." A86. But while those principles might have been "in mind," they were not reflected in the PTAB's substantive analysis. The PTAB next offered one paragraph noting that Woodhill had "no deficiencies to cure," three pages discussing whether a person of ordinary skill would have combined Woodhill with an MD5 hash function, and two pages addressing licenses as a secondary consideration. A86-91. In other words, there was no substantial discussion regarding the scope and content of the prior art, no substantial discussion regarding the differences between the prior art and the claims at issue (aside from that apparently relating to the single "identity means" element), and no discussion at all regarding the level of ordinary skill in the art. A86-91.

The lack of findings with respect to the relevant level of skill is particularly troubling, as this "determination … is an integral part of the *Graham* analysis." *Ruiz v. A.B. Chance Co.*, 234 F.3d 654, 666 (Fed. Cir. 2000). Without an express level-of-ordinary-skill finding, a tribunal "cannot properly assess obviousness because the critical question is whether a claimed invention would have been obvious at the time it was made to one with ordinary skill in the art." *Custom Accessories*, 807 F.2d at 962. Two of the PTAB's final written decisions in these consolidated appeals briefly make note—though not as a part of any obviousness analysis—of PersonalWeb's position regarding the level of ordinary skill in the art. A46; A119; A140-43. The remaining four final written decisions contain no discussion whatsoever of the level of ordinary skill. A222-25; A310-42; A394-404;

A463-82. And nowhere does the PTAB make any express finding regarding the level of ordinary skill in the art—a critical flaw in its analysis. *Ruiz*, 234 F.3d at 666-67; *see also Custom Accessories*, 807 F.2d at 963 (noting that the failure to make level-of-skill findings was "evidence that *Graham* was not in fact applied").

The conclusory nature of the PTAB's obviousness analysis, as well as the critical omission of sufficient factual findings under the four *Graham* factors, necessitates reversal of its obviousness determinations across the board in these consolidated appeals. *Custom Accessories*, 807 F.2d at 963; *Loctite*, 781 F.2d at 873; *Gechter*, 116 F.3d at 1460; *In re Bond*, 910 F.2d 831, 833-34 (Fed. Cir. 1990); *Florida Power & Light Co. v. Lorion*, 470 U.S. 729, 744 (1985).

VI. CONCLUSION AND RELIEF REQUESTED

For all of these reasons, this Court should reverse the PTAB and hold that all of the challenged claims are patentable. Alternatively, the Court should reverse the PTAB and remand for further proceedings under proper claim constructions and corrected applications of the law of anticipation and obviousness.

Date: November 12, 2014 Respectfully submitted,

/s/ Roderick G. Dorman

Roderick G. Dorman
Principal Counsel
Lawrence M. Hadley
McKOOL SMITH HENNIGAN, P.C.
865 South Figueroa Street, Suite 2900
Los Angeles, CA 90017
(213) 694-1200

Pierre J. Hubert
Joel L. Thollander
McKOOL SMITH, P.C.
300 W. 6th Street, Suite 1700
Austin, Texas 78701
(512) 692-8700

Daniel L. Geyser
McKOOL SMITH, P.C.
300 Crescent Court, Suite 1500
Dallas, TX 75201
(214) 978-4000

Attorneys for Appellant
PersonalWeb Technologies, LLC

ADDENDUM

Trials@uspto.gov
571-272-7822

Paper 83
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION and VMWARE, INC.,
Petitioners,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00082
Patent 5,978,791

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

ZECHER, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. BACKGROUND

EMC Corporation and VMware, Inc. (collectively, "EMC") filed a Petition on December 15, 2012, requesting an *inter partes* review of claims 1-4, 29-33, and 41 ("the challenged claims") of U.S. Patent No. 5,978,791 (Ex. 1001, "the '791 patent"). Paper 8 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") timely filed a Patent Owner's Preliminary Response. Paper 15 ("Prelim. Resp."). Taking into account PersonalWeb's Preliminary Response, the Board determined that the information presented in the Petition demonstrated that there was a reasonable likelihood that EMC would prevail in challenging claims 1-4, 29-33, and 41 as unpatentable under 35 U.S.C. §§ 102(e) and 103(a). Pursuant to 35 U.S.C. § 314, the Board instituted this proceeding on May 17, 2013, as to the challenged claims of the '791 patent. Paper 21 ("Dec.").

During this proceeding, PersonalWeb timely filed a Patent Owner Response (Paper 47, "PO Resp."), and EMC timely filed a Reply to the Patent Owner Response (Paper 55, "Reply"). A consolidated oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This decision is a final written decision under 35 U.S.C. § 318(a) as to the patentability of the challenged claims. Based on the record before us, EMC has demonstrated

[1] This proceeding, as well as IPR2013-00083, IPR2013-00084, IPR2013-00085, IPR2013-00086, and IPR2013-00087, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 82.

by a preponderance of the evidence that claims 1-4, 29-33, and 41 are unpatentable.

A. The Invention of the ’791 Patent

The invention of the ’791 patent relates to a data processing system that identifies data items using substantially unique identifiers, otherwise referred to as True Names, which depend on all the data in the data item and only on the data in the data item. Ex. 1001, 1:14-18, 3:29-32, and 6:6-10. According to the ’791 patent, the identity of a data item depends only on the data and is independent of the data item’s name, origin, location, address, or other information not directly derivable from the data associated therewith. Ex. 1001, 3:33-35. The invention of the ’791 patent also examines the identities of a plurality of data items in order to determine whether a particular data item is present in the data processing system. Ex. 1001, 3:36-39.

B. Illustrative Claims

Claims 1, 30, and 33 are independent claims. Claims 2-4 and 29 depend directly or indirectly from independent claim 1. Claims 31, 32, and 41 depend directly or indirectly from independent claim 30. Independent claims 1, 30, and 33 are illustrative of the invention of the ’791 patent and are reproduced below:

> 1. In a data processing system, an apparatus comprising:
>
> identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier*;* and

> existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

Ex. 1001, 39:14-23.

> 30. A method of identifying a data item present in a data processing system for subsequent access to the data item, the method comprising:
>
> determining a substantial unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and
>
> accessing a data item in the system using the identifier of the data item.

Ex. 1001, 42:58-67.

> 33. A method of duplicating a given data item present at a source location to a destination location in a data processing system, the method comprising:
>
> determining a substantially unique identifier for the given data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier;
>
> determining, using the data identifier, whether the data item is present at the destination location; and
>
> based on the determining whether the data item is present, providing the destination location with the data item only if the data item is not present at the destination.

Ex. 1001, 43:11-23.

C. Related Proceedings

EMC indicates that the ’791 patent was asserted against it in *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED, pending in the United States District Court for the

Eastern District of Texas. Pet. 1. EMC also filed five other petitions seeking *inter partes* review of the following patents: (1) U.S. Patent No. 6,415,280 (*EMC Corp. and VMware, Inc. v. PersonalWeb Techs., LLC,* IPR2013-00083); (2) U.S. Patent No. 7,945,544 (*EMC Corp. v. PersonalWeb Techs., LLC*, IPR2013-00084); (3) U.S. Patent No. 7,945,539 (*EMC Corp. v. PersonalWeb Techs., LLC*, IPR2013-00085); (4) U.S. Patent No. 7,949,662 (*EMC Corp. v. PersonalWeb Techs.*, LLC, IPR2013-00086); and (5) U.S. Patent No. 8,001,096 (*EMC Corp. v. PersonalWeb Techs., LLC,* IPR2013-00087). *Id*.

D. Prior Art Relied Upon

EMC relies upon the following prior art reference:

Woodhill	US 5,649,196	July 15, 1997 (effectively filed July 1, 1993)	Ex. 1005

E. Grounds of Unpatentability

We instituted this proceeding based on the grounds of unpatentability set forth in the table below.

Claims	**Basis**	**Reference**
1-4, 29-33, and 41	§ 102(e)	Woodhill
1-4 and 29	§ 103(a)	Woodhill

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, we construe a claim by applying the broadest reasonable interpretation in light of the specification of the patent in which it appears. 37 C.F.R. § 42.100(b); *see* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). Under the broadest reasonable interpretation standard, claim terms are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). We must be careful not to read limitations from a particular embodiment appearing in the specification into the claim if the claim language is broader than that embodiment. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993). If a feature in the disclosure is not necessary to give meaning to what the inventor means by a claim term, it would be "extraneous" and, therefore, should not be read into the claim. *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1249 (Fed. Cir. 1998); *E.I. du Pont de Nemours & Co. v. Phillips Petroleum Co*., 849 F.2d 1430, 1433 (Fed. Cir. 1988).

In its Petition, EMC identified five claim terms and provided a claim construction for those terms. Pet. 4-6. Those claim terms are listed as follows: (1) "substantially unique identifier"; (2) "using the identifier"; (3) "data" and "data item"; (4) "location"; and (5) "True Name, data identity, and data identifier." *Id*. In the Decision to Institute, we construed each claim term identified by EMC. Dec. 13-16.

In its Petition, EMC also identified several means-plus-function limitations that invoke 35 U.S.C. § 112, ¶ 6, and their corresponding structure for performing the claimed function. Pet. 6-8. Those means-plus-function limitations are listed as follows: (1) "identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier"; (2) "existence means for determining whether a particular item is present in the system, by examining the identifiers of the plurality of data items"; (3) "local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item"; (4) "data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item"; and (5) "access means for accessing a particular data item using the identifier of the data item." In the Decision to Institute, we construed each means-plus-function limitation identified by EMC to cover the corresponding structure described in the specification of the '791 patent and equivalents thereof. Dec. 17-26, *see* 35 U.S.C. § 112, ¶ 6.

With one exception, PersonalWeb agrees with our claim constructions in the Decision to Institute. PO Resp. 1-3 (quoting Dec. 13-16, 20-25). PersonalWeb proposes an alternative claim construction for the following means-plus-function limitation recited in independent claim 1:

> identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.

Id. at 13-14 (quoting Ex. 1001, 39:16-21).

We will address PersonalWeb's alternative claim construction for this means-plus-function limitation below.

> *1.* "*identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier" (Claim 1)*

As we indicated in the Decision to Institute, both parties agreed that the claimed function of this means-plus-function limitation is "determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier." Dec. 18 (quoting Ex. 1001, 39:16-21). We then identified a data processor programmed to perform a hash function, e.g., MD5 or SHA, as the corresponding structure for performing the claimed function. *Id*. at 20.

In its Patent Owner Response, PersonalWeb contends that our construction of the corresponding structure for performing the claimed function is overly broad. PO Resp. 14. PersonalWeb argues that the specification of the '791 patent discloses that the identity calculating mechanism "must" have at least five properties, and "must" be employed on a system wide basis. *Id*. (citing Ex. 1001, 12:61-13:9, 13:15-19). PersonalWeb further argues that the specification of the '791 patent describes corresponding structure that is necessary to determine a substantially unique identifier "for any of a plurality of data items present in the system," as claimed. *Id*. (citing Ex. 1001, 14:12-31). PersonalWeb asserts that our claim construction in the Decision to Institute does not reflect the aforementioned features disclosed in the specification of the '791 patent that are necessary to perform the claim function. *Id*. Based on those arguments, PersonalWeb identifies at least one processor programmed to perform the Calculate True name mechanism as the corresponding structure for performing the claimed function. *Id*. (citing Ex. 1001, 7:62-63, 12:54-13:19, 14:1-39).

To the extent that PersonalWeb argues that we failed to consider the use of the term "must" in the specification of the '791 patent when we previously construed the corresponding structure that performs the claimed function, we disagree. As we explained above, PersonalWeb directs us to the disclosure in the specification of the '791 patent that states, "[t]he function MD *must* have the following properties . . . [and] [t]hese functions (or algorithms) include MD4, MD5, and SHA." Ex. 1001, 12:61-13:14 (emphasis added). PersonalWeb also directs us to the disclosure in the specification of the '791 patent that states "[i]n the presently preferred embodiments, either MD5 or SHA is employed as the basis for the computation of True Names. Whichever of these two message digest functions is employed, the same function *must* be employed on a system-wide basis." Ex. 1001, 13:15-19 (emphasis added).

Although the specification of the '791 patent uses the absolute term "must" when describing MD hash functions generally, and hash functions MD5 and SHA specifically, it nonetheless describes these hash functions in the context of "preferred embodiments." Therefore, we included the hash functions MD5 and SHA in our claim construction as examples only. Dec. 18-20. Our determination in that regard incorporates the features necessary to perform the claimed function, yet does not conflict with the principle that this means-plus-function limitation is to be given its broadest reasonable interpretation. *In re Donaldson*, 16 F.3d 1189, 1194 (Fed. Cir. 1994). In other words, we did not view hash functions MD5 and SHA as "necessary" to perform the claimed function because they were part of the preferred embodiments disclosed in the specification of the '791 patent. Claim interpretation under § 112, ¶ 6, does not "permit incorporation of structure from the written description beyond that necessary to perform the claimed function." *Micro Chem., Inc. v. Great Plains Chem. Co.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999).

We also are not persuaded by PersonalWeb's argument that the specification of the '791 patent describes corresponding structure that is necessary to determine a substantially unique identifier "for any of a plurality of data items present in the system," as claimed. The alleged corresponding structure referenced in PersonalWeb's argument is the embodiment illustrated in Figure 10(b), which is a flowchart depicting the operations associated with calculating the True Name of an arbitrary, i.e., simple or compound, data item. Ex. 1001, 14:1-3, 13-15. PersonalWeb does not explain adequately why the steps illustrated in that embodiment are necessary to perform the claimed function, nor does PersonalWeb explain why such steps must be part of the algorithm that provides the necessary structure under § 112, ¶ 6. We do not find the steps illustrated in Figure 10(b) as "necessary" to perform the claimed function and, therefore, such steps should not be read into the corresponding structure for performing the claimed function. *See Micro Chem.*, 194 F.3d at 1258.

Applying the broadest reasonable interpretation standard, we maintain that the corresponding structure identified in the specification of the '791 patent for performing the claimed function of "determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier" is a data processor programmed to perform a hash function, e.g., MD5 or SHA.

B. The Level of Ordinary Skill in the Art

In determining the level of one with ordinary skill in the art, we note that various factors may be considered, including "type of problems encountered in the art; prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field." *In re GPAC*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (citing *Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986)). There is sufficient evidence in the record before us that reflects the knowledge level of a person with ordinary skill in the art. PersonalWeb's expert, Dr. Robert B.K. Dewar, attests that a person with ordinary skill in the art would be an individual with a bachelor's degree in computer science who possesses ten to fifteen years of teaching or work experience in the field of data processing systems. Ex. 2013 ¶ 18.

C. Anticipation by Woodhill—Claims 1-4, 29-33, and 41

EMC contends that claims 1-4, 29-33, and 41 are anticipated under § 102(e) by Woodhill. Pet. 51-59. In support of that alleged ground of unpatentability, EMC provides explanations as to how Woodhill describes each claim limitation. *Id*. (citing Ex. 1041). EMC also submits the declarations of Dr. Douglas W. Clark (Ex. 1009 ¶¶ 81-95; Ex. 1081) to support its positions. Upon reviewing EMC's Petition and supporting evidence, as well as PersonalWeb's Patent Owner Response and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claims 1-4, 29-33, and 41 are anticipated by Woodhill.

We begin our analysis with the principles of law that generally apply to a ground of unpatentability based on anticipation, followed by a brief discussion of Woodhill, and then we turn to the arguments presented by both EMC and PersonalWeb that are directed towards each challenged claim.

1. Principles of Law

To establish anticipation under § 102(e), "all of the elements and limitations of the claim must be shown in a single prior reference, arranged as in the claim." *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001). "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987). We analyze the ground of unpatentability based on anticipation by Woodhill with the above-stated principles in mind.

2. Woodhill

Woodhill generally relates to a system and method for distributed storage management on a networked computer system that includes a remote backup file server in communication with one or more local area networks. Ex. 1005, 1:11-17. Figure 1 of Woodhill, which is reproduced below, illustrates networked computer system 10. Ex. 1005, 2:56-58.

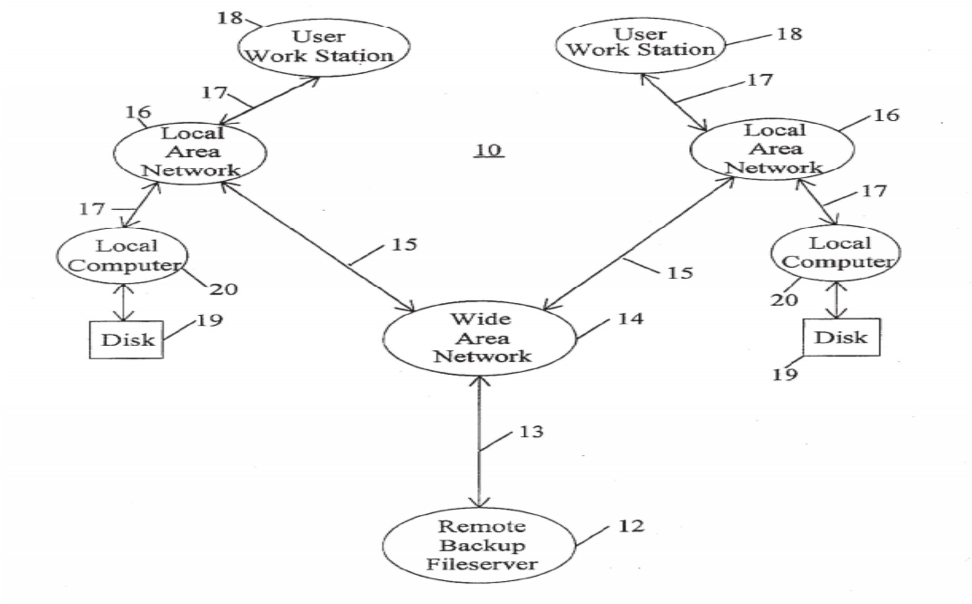


FIG. 1

As shown in Figure 1 of Woodhill, remote backup file server 12 communicates with wide area network 14 via data path 13, wide area network 14 communicates with a plurality of local area networks 16 via data paths 15, and each local area network 16 communicates with multiple user workstations 18 and local computers 20 via data paths 17. Ex. 1005, 3:12-31. The storage space on each disk drive 19 on each local computer 20 is allocated according to the hierarchy illustrated in Figure 2. Ex. 1005, 3:31-44.

Figure 2 of Woodhill, which is reproduced below, illustrates Distributed Storage Manager program 24 that allocates storage space on each of the storage devices in networked computer system 10. Ex. 1005, 2:59-62.

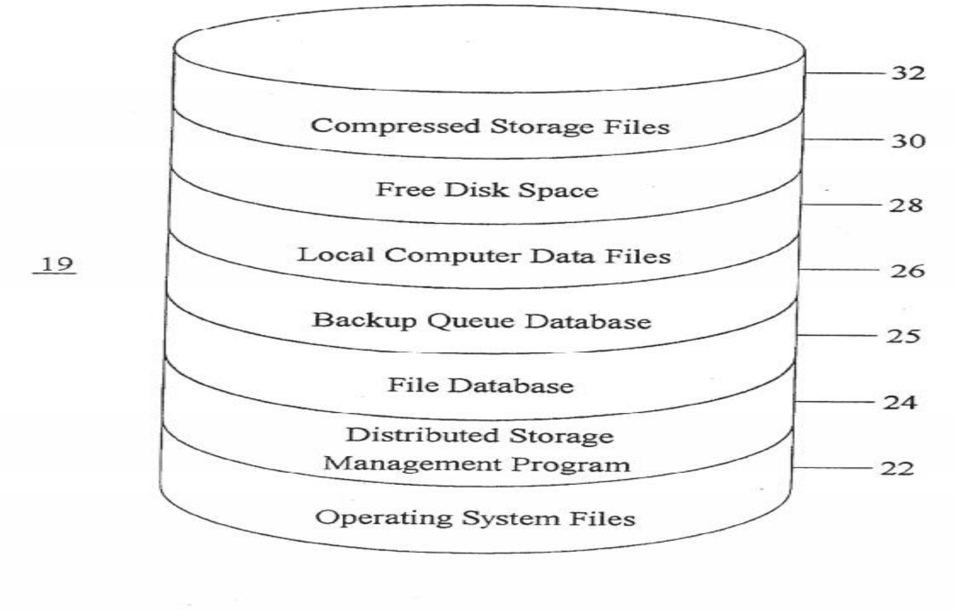


FIG. 2

As shown in Figure 2 of Woodhill, Distributed Storage Manager program 24 builds and maintains File Database 25 on the one or more disk drives 19 on each local computer 20 in networked computer system 10. Ex. 1005, 3:45-49. Distributed Storage Manager program 24 views a file as a collection of data streams. Ex. 1005, 4:13-15. Woodhill defines a data stream as a distinct collection of data within a file that may change independently from other distinct collections of data within the file. Ex. 1005, 4:15-18. For instance, Woodhill discloses that a file may contain both its normal data and any extended attribute data. Ex. 1004, 4:18-19. Depending on the size of the data stream, Distributed Storage Manager program 24 divides each data stream into one or more binary objects. Ex. 1005, 4:21-30.

Figure 3 of Woodhill, which is reproduced below, illustrates File Database 25 used by Distributed Storage Manager program 24. Ex. 1005, 2:63-64.

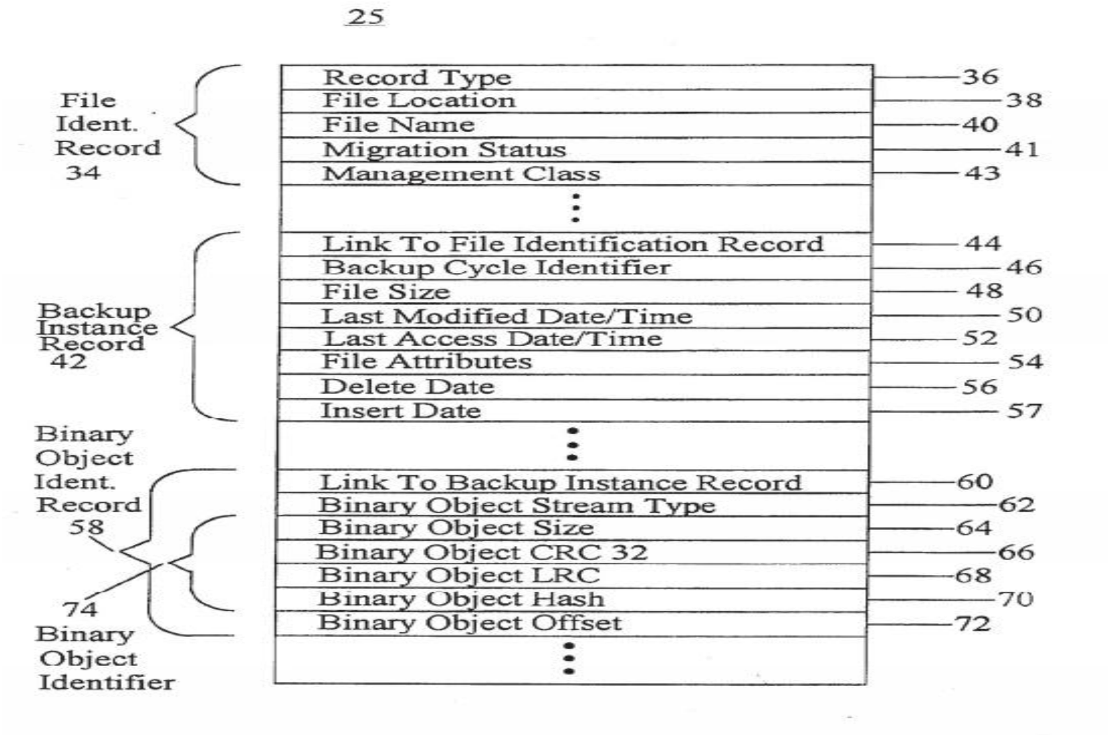


FIG. 3

As shown in Figure 3 of Woodhill, File Database 25 includes the following three levels of records organized according to a predefined hierarchy: (1) File Identification Record 34; (2) Backup Instance Record 42; and (3) Binary Object Identification Record 58. Ex. 1005, 3:54-4:47. Binary Object Identification Record 58 includes, amongst other things, Binary Object Identifier 74 that comprises Binary Object Size 64, Binary Object CRC32 66, Binary Object LRC 68, and Binary Object Hash 70. Ex. 1005, 4:45-47, 7:64-8:1. Binary Object Identifier 74 is a unique identifier for each binary object that is backed up. Ex. 1005, 4:45-47.

Although Woodhill discloses calculating Binary Object Identifier 74 in various ways, e.g., using a binary hash algorithm (Ex.1005, 8:1-31), the key notion is that Binary Object Identifier 74 is calculated from the content of the data instead of from an external or arbitrary source. Ex. 1005, 8:38-

42. In other words, Woodhill recognizes that the critical feature in creating Binary Object Identifier 74 is that the identifier should be based on the contents of the binary object, such that Binary Object Identifier 74 changes when the contents of the binary object changes. Ex. 1005, 8:58-62. Therefore, duplicate binary objects, even if resident on different types of computers in the network, may be recognized by their identical Binary Object Identifiers 74. Ex. 1005, 8:62-65.

Woodhill discloses that Distributed Storage Manager program 24 performs two backup operations concurrently. Ex. 1005, 9:30-31. First, Distributed Storage Manager program 24 stores a compressed copy of each binary object that it needs to restore disk drive 19 on each local computer 20 somewhere on local area network 16 other than on local computer 20 where the binary object originally resided. Ex. 1005, 9:31-36. Second, Distributed Storage Manager program 24 transmits new or changed binary objects to remote backup file server 12. Ex. 1005, 9:36-38.

Woodhill also discloses that Distributed Storage Manager program 24 performs auditing and reporting functions on a periodic basis to ensure that binary objects, which already have been backed up, may be restored. Ex. 1005, 18:11-13. Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by a Binary Object Identification Record 58 stored in File Database 25. Ex. 1005, 18:16-19.

3. Claim 1

a. "determining whether a particular data item is present in the system"

Independent claim 1 recites, in relevant part, an "existence means for *determining whether a particular data item is present in the system*, by examining the identifiers of the plurality of data items." Ex. 1001, 39:21-23 (emphasis added).

In its Petition, EMC contends that Woodhill determines the existence of a binary object by examining Binary Object Identifiers 74 on local computers 20, which are part of local area networks 16, and on remote backup file server 12. Pet. 57 (citing Ex. 1009 ¶¶ 85-87; Ex. 1005, 8:62-9:23). For example, EMC argues that Woodhill uses Binary Object Identifier 74 to check whether a binary object has changed since it was last backed up, as well as to check whether a local copy of a binary object is available to be restored. *Id*. (citing Ex. 1009 ¶ 86; Ex. 1005, 9:14-22).

In its Patent Owner Response, PersonalWeb contends that Woodhill only determines whether a particular Binary Object Identifier 74 for a binary object is present for the most-recently backed up version of a single file, and therefore, Woodhill cannot "determine[] whether a particular data item is present in the system," as claimed. PO Resp. 4-7. In particular, PersonalWeb argues that the ability to determine whether a particular file is present in the system requires the ability to look at information for more than one file. *Id*. at 5. PersonalWeb further argues that one would need the ability to look at information regarding all the files in a system in order to determine if a particular file is present in the system. *Id*. PersonalWeb alleges that Woodhill cannot determine whether a particular binary object is present in its system because it is incapable of determining if that particular binary object is present in any of the thousands, if not millions, of files in the system. *Id*. at 8. PersonalWeb relies upon the declaration of Robert B.K. Dewar to support its positions. Ex. 2013 ¶¶ 21-28.

In its Reply, EMC contends that PersonalWeb's argument is predicated on the notion that one would need to have the ability to look at all the files in a system in order to determine whether a file is present in the system. Reply 1-2. EMC argues that independent claim 1 is not that specific, but instead it generally refers to determining whether a data item is present in the system. *Id*. at 2. EMC directs our attention to related district court litigation, where the court was not persuaded by a similar argument presented by PersonalWeb. *Id*. at 2 (citing Ex. 1074, 35). We are not persuaded by PersonalWeb's arguments because they are based on an overly narrow claim construction.

As we explained in the Decision to Institute, we identified the corresponding structure for performing the claimed function of "determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items" to be a data processor programmed according to step S232 illustrated in Figure 11 or step S260 illustrated in Figure 14. Dec. 20-22. With respect to step S232 illustrated in Figure 11, the specification of the '791 patent discloses "look[ing] for an entry for the True Name in the True File registry 126 (Step S232) and determin[ing] whether a True Name entry, record 140, exists in the True file registry 126." Ex. 1001, 14:53-56. With respect to step S260 illustrated in Figure 14 of the '791 patent, the specification of the '791 patent discloses, "if desired, confirm[ing] that the True Name exists locally by searching for it in the True Name registry or local directory extensions table 135 (Step S260)." Ex. 1001, 15:54-56.

Contrary to PersonalWeb's arguments, the claimed function of "determining whether a particular data item is present in the system" does not encompass searching all the files in a system. Instead, according to the specification of the ’791 patent, it simply includes determining whether a file exists in a registry or table. *See, e.g.*, Ex. 1001, 14:53-56, 15:54-56, fig. 11, step S232, fig. 14, step S260. Woodhill recognizes duplicate binary objects residing on different types of computers in the network by their identical Binary Object Identifiers 74. Ex. 1005, 8:62-65. During Woodhill’s backup procedure, Binary Object Identifiers 74 are calculated for each binary object and then compared against their counterparts in File Database 25. Ex. 1005, Ex. 9:14-16. For example, Woodhill discloses that Distributed Storage Manager program 24 compares a newly calculated Binary Object Identifier 74 for a particular binary object with Binary Object Identifier 74 associated with the most recent version of that binary object. Ex. 1005, 9:16-22. Dr. Clark testifies that this comparison of Binary Object Identifiers 74 is just one relevant example of determining whether and where a particular binary object is present in its system. Ex. 1009 ¶ 86.

In summary, we agree with EMC that Woodhill’s backup procedure, which includes calculating Binary Object Identifiers 74 for each binary object and then comparing them against their counterparts in File Database 25, describes the function of “determining whether a particular data item is present in the system,” as recited in independent claim 1.

b. “existence means”

Independent claim 1 recites, in relevant part, an “*existence means* for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” Ex. 1001, 39:21-23 (emphasis added).

In its Patent Owner Response, PersonalWeb relies upon essentially the same argument presented above with respect to the claimed function of "determining whether a particular data item is present in the system." *Compare* PO Resp. 4-10 *with* PO Resp. 10-13. That is, PersonalWeb contends that Woodhill does not search for the newly calculated Binary Object Identifier 74 in a registry or table that includes a plurality of Binary Object Identifiers 74 associated with different files in the system. *Id*. at 11-12. Therefore, PersonalWeb asserts that Woodhill does not disclose a structure tantamount to the corresponding structure for the claimed "existence means" because Woodhill does not perform the identical function in substantially the same way to achieve substantially the same results. *Id*.

In its Reply, EMC contends that Woodhill's File Database 25 is equivalent to True file registry 126 described in the specification of the '791 patent. Reply 4 (citing Ex. 1001, 9:36-67). EMC reiterates that we properly identified the corresponding structure for the "existence means" in the Decision to Institute (Dec. 20-22), and then contends that both the '791 patent and our construction simply require confirming whether an identifier exists in a database that has a plurality of identifiers. Reply 4-5.

As we explained above, the claimed function associated with the "existence means" simply encompasses determining whether a file exists in a registry or table. *See, e.g.*, Ex. 1001, 14:53-56, 15:54-56, fig. 11, step S232, fig. 14, step S260. It does not include searching all the files in a system. According to Woodhill, both its system and method for managing storage space on network computer system 10 include comparing the current value of the binary object identifier associated with a particular binary object to *one or more* previous values of the binary object identifier associated with that particular binary object. Ex. 1005, 2:14-17, 33-36 (emphasis added); *see also* Ex. 2007, 22.[2] Independent claim 1 of Woodhill further recites, in relevant part, "means for comparing said current name of a particular binary object to *one or more* previous names of said binary object." Ex. 1005, 22:5-7 (emphasis added); *see also* Ex. 2007, 62 (originally presented independent claim 1). In our view, these disclosures in Woodhill apply to its backup procedure and, in particular, support a finding that File Database 25 stores a plurality of Binary Object Identifiers 74 associated with different binary objects or files that have been backed up in the system (*see, e.g.*, Ex. 1005, 3:49-52, 4:30-34, 9:8-22). *Cf. In re Preda*, 401 F.2d 825, 826 (CCPA 1968) ("[I]t is proper to take into account not only specific teachings of the references but also the inferences which one skilled in the art would reasonably be expected to draw therefrom.").

As EMC explains in its Reply, Dr. Clark testifies that Woodhill determines whether a binary object or file is present in the system by confirming that its Binary Object Identifier 74 already exists among the plurality of Binary Object Identifiers 74 stored in File Database 25. Reply 5 (citing Ex. 1081 ¶ 14). We credit Dr. Clark's testimony in that regard because it is consistent with Woodhill's summary of its own invention, the broader disclosure provided by independent claim 1, and the description of File Database 25.

[2] Exhibit 2007 includes excerpts from the file history of Woodhill. PersonalWeb did not provide any page numbers for this Exhibit. For purposes of this decision, page 1 is the page that includes "Exhibit PersonalWeb 2007" in the lower, right-hand corner. The remaining pages are numbered consecutively therefrom.

In light of our analysis above, we agree with EMC that Woodhill discloses a structure equivalent to the corresponding structure for the claimed "existence means" because it performs an identical function in substantially the same way to achieve substantially the same results. *See, e.g.*, *Odetics, Inc. v. Storage Tech. Corp.*, 185 F.3d 1259, 1267 (Fed. Cir. 1999). In other words, Woodhill's backup procedure, which includes calculating Binary Object Identifiers 74 for each binary object and then comparing them against a plurality of Binary Object Identifiers 74 stored in File Database 25, describes the function of "determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items," as recited in independent claim 1.

c. "identity means"

Independent claim 1 recites, in relevant part,

> *Identity means* for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.

Ex. 1001, 39:16-21.

In its Petition, EMC contends that Woodhill's Binary Object Identifiers 74 constitute the claimed "substantially unique identifiers" determined using the contents in the binary object. Pet. 56-57 (citing Ex. 1005, 7:60-8:1, fig. 3; Ex. 1009 ¶¶ 83-84). EMC argues that, when calculating the Binary Object Identifiers 74, Woodhill uses "all of" the data of a binary object and "only" the data of the binary object. *Id*. at 57 (citing Ex. 1009 ¶ 84; Ex. 1005, 8:1-31). EMC argues that two identical binary objects in Woodhill's system will have the same Binary Object Identifier 74 because each Binary Object Identifier 74 is based on the data of the binary object associated therewith. *Id*.

In its Patent Owner Response, PersonalWeb proposes an alternative claim construction for the claimed "identity means," and then contends that Woodhill fails to disclose the corresponding structure identified in its alternative claim construction. *Id*. at 14-15. In particular, PersonalWeb argues that, when the '791 patent determines whether a data item is compound, the claimed "identity means" requires a cryptographic hash of cryptographic hashes ("a hash of hashes"). *Id*. PersonalWeb argues that, although Woodhill discloses calculating Binary Object Identifier 74 for a binary object by applying a hash function to the binary object, it does not apply a hash function to Binary Object Identifier 74, itself. *Id*. In addition, PersonalWeb contends that Woodhill does not disclose a cryptographic hash, such as MD5, SHA, or anything equivalent thereto. *Id*.

As we explained previously, we disagree with the alternative claim construction for "identity means" proposed by PersonalWeb in its Patent Owner Response. To the extent PersonalWeb now argues that the claimed "identity means" requires a hash of hashes, we also disagree. Similar to our explanation in the Decision to Institute, PersonalWeb's argument in that regard is not commensurate in scope with our claim construction of "identity means." *See* Dec. 27-28. The corresponding structure for performing the claimed function of "determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier" is a data processor programmed to perform a hash function, e.g., MD5 or SHA. Neither the specification of the '791 patent, nor the claim itself, indicates that the "identity means" requires determining a substantial unique identifier for a compound data item, much less using a hash of hashes when determining whether a data item is compound.

We also are not persuaded by PersonalWeb's argument that Woodhill does not disclose a cryptographic hash, such as an MD5, SHA, or anything equivalent thereto. Woodhill discloses various ways to calculate Binary Object Identifier 74 for a particular binary object, including using a binary hash algorithm. Ex.1005, 8:1-31. The key notion in Woodhill is that Binary Object Identifier 74 is calculated based on the content of each binary object instead of from an external or arbitrary source. Ex. 1005, 8:38-42. In other words, Woodhill creates Binary Object Identifier 74 for a binary object based on the contents of the binary object, such that Binary Object Identifier 74 changes when the contents of the binary object changes. Ex. 1005, 8:58-62. Based on these cited disclosures, Woodhill's binary hash algorithm relies on "all of" the data of a binary object and "only" the data of the binary object when calculating Binary Object Identifier 74.

In summary, we agree with EMC that Woodhill's disclosure of calculating Binary Object Identifier 74 for a particular binary object describes the corresponding structure for performing the claimed function associated with the "identity means," as recited in independent claim 1. For the foregoing reasons, we conclude that that EMC has demonstrated by a preponderance of the evidence that independent claim 1 is anticipated by Woodhill.

4. Claim 2

a. "determining whether . . . a particular data item is present at a particular location in the system"

Dependent claim 2 recites a "local existence means for *determining whether* an instance of *a particular data item is present at a particular location in the system*, based on the identifier of the data item." Ex. 1001, 39:25-29 (emphasis added).

In its Petition, EMC contends that Woodhill discloses a "local existence means," as claimed. Pet. 58 (citing Ex. 1005, 9:5-23). In particular, EMC argues that Woodhill's remote backup server 12 constitutes the claimed "particular location in the system." Ex. 1041, 15. EMC also offers the testimony of Dr. Clark to support its position. Ex. 1009 ¶¶ 88, 89.

In its Patent Owner Response, PersonalWeb presents essentially the same arguments discussed above with respect to independent claim 1. PO Resp. 16-17. That is, PersonalWeb argues that Woodhill only determines whether Binary Object Identifier 74 for a particular binary object is present for the most-recent version of a particular file at remote backup server 12, and does not determine whether that particular file is present in the many other files stored at remote backup server 12. *Id*.

As we explained in the Decision to Institute, we identified the corresponding structure for performing the claimed function of "determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item" to be a data processor programmed according to step S260 illustrated in Figure 14. Dec. 21-22. With respect to step S260 illustrated in Figure 14, the specification of the '791 patent discloses, "if desired, confirm[ing] that the

True Name exists locally by searching for it in the True Name registry or local directory extensions table 135 (Step S260)." Ex. 1001, 15:54-56.

Contrary to PersonalWeb's arguments, the claimed function of "determining whether . . . a particular data item is present at a particular location in the system" does not encompass searching all the files in a system. Instead, according to the specification of the '791 patent, it simply includes determining whether a file exists in a registry or table. *See, e.g.*, Ex. 1001, 15:54-56, fig. 14, step S260.

To support its position regarding dependent claim 2, EMC, once again, directs our attention to Woodhill's backup procedure. During Woodhill's backup procedure, Distributed Storage Manager program 24 determines whether a particular binary object has changed using the version of the binary object that previously was backed up. Ex. 1005, 9:6-9. Dr. Clark testifies that File Database 25 contains a list of Binary Object Identifiers 74 for binary objects recently backed up and stored in the system, including the binary objects backed up and stored in remote backup file server 12. Ex. 1009 ¶ 89. Dr. Clark also testifies that, when comparing Binary Object Identifier 74 calculated during the current backup cycle with those stored in File Database 25, Distributed Storage Manager program 24 essentially determines the existence, at remote backup file server 12, of the particular binary object being processed. Ex. 1009 ¶ 89; *see also* Ex. 1081 ¶¶ 5-8. We credit Dr. Clark's testimony because it is consistent with Woodhill's description of the backup procedure.

PersonalWeb further contends that, although it agrees with our claim construction of the claim term "location," a single file in Woodhill does not constitute the claimed "location." PO Resp. 17 (citing Dec. 15-16; Ex. 2013

¶ 39). PersonalWeb's argument that a single file in Woodhill does not constitute the claimed "particular location in the system" is misplaced. There is no indication in the record before us that EMC takes the position that Woodhill's disclosure of a single file constitutes the claimed "particular location in the system." Instead, as explained above, EMC takes the position that Woodhill's disclosure of remote backup server 12 constitutes the claimed "particular location in the system." Ex. 1041, 15.

In summary, we agree with EMC that Woodhill's backup procedure, which includes examining Binary Object Identifiers 74 stored in File database 25 to determine if the most recent version of a binary object is present at remote backup file server 12, describes the function of "determining whether . . . a particular data item is present at a particular location in the system," as recited in dependent claim 2.

b. "local existence means"

In its Patent Owner Response, PersonalWeb relies upon essentially the same arguments presented above with respect the claimed function of this means-plus-function limitation, "determining whether . . . a particular data item is present at a particular location in the system," to rebut EMC explanations as to how Woodhill describes the claimed "local existence means." *Compare* PO Resp. 16-18 *with* PO Resp. 18-20. That is, PersonalWeb contends that Woodhill does not search for the newly calculated Binary Object Identifier 74 in a registry or table that includes a plurality of Binary Object Identifiers 74 associated with different files in the system. PO Resp. 19. Therefore, PersonalWeb asserts that Woodhill does not disclose a structure tantamount to the corresponding structure for the claimed "local existence means" because Woodhill does not perform the identical function in substantially the same way to achieve substantially the same results. *Id*.

As we explained above, the claimed function associated with the "local existence means" simply encompasses determining whether a file exists in a registry or table. *See, e.g.*, Ex. 1001, 15:54-56, fig. 14, step S260. It does not include searching all the files in a system. Moreover, Woodhill discloses that File Database 25 stores a plurality of Binary Object Identifiers 74 associated with different binary objects or files that have been backed up in the system. *See, e.g.*, Ex. 1005, 3:49-52, 4:30-34, 9:8-22. Dr. Clark testifies that Woodhill determines whether a binary object or file is present at remote backup file server 12 by confirming that its Binary Object Identifier 74 already exists among the plurality of Binary Object Identifiers 74 stored in File Database 25. Ex. 1009 ¶ 89; *see also* Ex. 1081 ¶ 14. We credit Dr. Clark's testimony in that regard because it is consistent with Woodhill's description of File Database 25, as well as its description of the backup procedure.

In light of our analysis above, we agree with EMC that Woodhill discloses a structure tantamount to the corresponding structure for the claimed "local existence means" because it performs an identical function in substantially the same way to achieve substantially the same results. *See, e.g.*, *Odetics*, 185 F.3d at 1267. In other words, Woodhill's backup procedure, which includes examining Binary Object Identifiers 74 stored in File database 25 to determine if the most recent version of a binary object is present at remote backup file server 12, describes the function of "determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item," as recited in dependent claim 2. For the foregoing reasons, we conclude that that EMC has demonstrated by a preponderance of the evidence that dependent claim 2 is anticipated by Woodhill.

5. Claim 3

a. "examining the identifiers of the plurality of data items at said particular location in the system"

Dependent claim 3 recites, in relevant part, "wherein said local existence means for determining whether a particular data item is present at a particular location in the system by *examining the identifiers of the plurality of data items at said particular location in the system*." Ex. 1001, 39:31-35 (emphasis added).

In its Petition, EMC contends that Woodhill discloses the "local existence means," as claimed. Pet. 58 (citing Ex. 1005, 9:5-23). In particular, EMC argues that Woodhill's Distributed Storage Manager program 24, which executes on a computer, accesses and checks Binary Object Identification Records 58 in File Database 25 to determine whether a local copy of a particular binary object is present on the local system before restoring a remote copy. Ex. 1041, 15. According to EMC, Woodhill's Distributed Storage Manager program 24 performs this function by examining Binary Object Identifiers 74 for the plurality of binary objects stored in File Database 25. *Id.* EMC also offers the testimony of Dr. Clark to support its position. Ex. 1009 ¶¶ 90, 91.

In its Patent Owner Response, PersonalWeb reiterates that, during Woodhill's backup procedure, Distributed Storage Manager program 24 compares Binary Object Identifier 74 for a newly processed binary object with only a single prior version of Binary Object Identifier 74. PO Resp. 21-

22. PersonalWeb maintains that Woodhill does not compare the newly created Binary Object Identifier 74 for a binary object or file with a plurality of previous Binary Object Identifiers 74 for that file. *Id*. at 21. Once again, we are not persuaded by PersonalWeb's argument.

Woodhill discloses that File Database 25 stores a plurality of Binary Object Identifiers 74 associated with different binary objects or files that have been backed up in the system. *See, e.g.*, Ex. 1005, 3:49-52, 4:30-34, 9:8-22. We agree with EMC that Woodhill's Distributed Storage Manager program 24 determines whether a binary object or file is present at remote backup file server 12 by examining Binary Object Identifiers 74 for the plurality of binary objects or files stored in File Database 25. Ex. 1041, 15. Dr. Clark further confirms EMC's position in that regard. Ex. 1009 ¶ 91; *see* Ex. 1008 ¶ 14. We credit Dr. Clark's testimony because it is consistent with Woodhill's description of File Database 25, as well as its description of the backup procedure.

In summary, we agree with EMC that Woodhill's backup procedure, which includes examining Binary Object Identifiers 74 stored in File database 25 to determine if a particular binary object is present at remote backup file server 12, describes the function of "examining the identifiers of the plurality of data items at said particular location in the system," as recited in dependent claim 3. For the foregoing reasons, we conclude that that EMC has demonstrated by a preponderance of the evidence that dependent claim 3 is anticipated by Woodhill.

6. Claim 4

a. "accessing a particular data item using the identifier for the data item"

Dependent claim 4 recites, in relevant part, "access means for *accessing a particular data item using the identifier of the data item*." Ex. 1001, 39:40-41 (emphasis added).

In its Petition, EMC contends that Woodhill discloses the "access means," as claimed. Pet. 58 (citing Ex. 1005, 7:60-8:65, 18:11-23). In particular, EMC argues that Woodhill's Distributed Storage Manager program 24 executes a self-audit procedure on a computer that accesses binary objects using their Binary Object Identifiers 74 during the backup/restore routine. Ex. 1041, 16. EMC also argues that Woodhill's Distributed Storage Manager program 24 performs self-audits by initiating a restore of a randomly selected binary object using its Binary Object Identification record 58, which includes, amongst other things, Binary Object Identifier 74. *Id*. EMC offers the testimony of Dr. Clark to support its position. Ex. 1009 ¶¶ 94, 95.

In its Patent Owner Response, PersonalWeb contends that Woodhill does not use Binary Object Identifier 74, which is part of Binary Object Identification record 58, to access a particular binary object. PO Resp. 24-29. In particular, PersonalWeb argues that, during Woodhill's self-audit procedure, Binary Object Identifier 74 is used merely for comparison purposes after the particular binary object already has been accessed to determine if the audit restore worked properly. *Id*. at 25 (citing Ex. 1005, 18:28-38; Ex. 2013 ¶ 54). PesonalWeb further argues that Woodhill's File Location 38 and File Name 40 in File Identification Record 34 are used to access a file containing a particular binary object, whereas Binary Object

Stream Type 62 and Binary Object Offset 72 in Binary Object Identification record 58 are used to locate the binary object in that file. *Id*. at 25-26 (Ex. 1005, 9:18-20, fig. 2; Ex. 2013 ¶ 55).

In its Reply, EMC contends that, contrary to PersonalWeb's arguments, Woodhill uses Binary Object Identifier 74 to name and restore binary objects. Reply 8 (citing Ex. 1005, 18:13-19, 22:3-4; Ex. 1081 ¶¶ 20-25). EMC directs us to Dr. Clark's testimony that there was no need to explain which subfields of Binary Object Identification Record 58 are used to access a binary object because, for such a basic and well known operation, a person of ordinary skill in the art would have understood that Binary Object Identifier 74 is used to look up a binary object. *Id*. at 9 (citing Ex. 1081 ¶ 20). EMC asserts this is why Woodhill's Binary Object Identifier 74 is referred as an "identifier," and why independent claim 1 of Woodhill refers to it as the "name" of a binary object. *Id*. EMC further contends that, instead of using Woodhill's File Location 38 and File Name 40 in File Identification Record 34 to access a binary object from remote backup file server 12, Woodhill uses Binary Object Identifier 74—the key component of Binary Object Identification record 58—to access the binary object from remote backup file server 12. *Id*. at 9-10.

As we explained in the Decision to Institute (Dec. 29), Woodhill discloses that Distributed Storage Manager program 24 performs auditing and reporting functions on a periodic basis in order to ensure that the binary objects, which already have been backed up, may be restored. Ex. 1005, 18:11-13. According to Woodhill, Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by Binary Object Identification Record 58 stored in File Database 25. Ex. 1005, 18:16-

19. Binary Object Identification Record 58 includes, amongst other things, a Binary Object Identifier 74, which is a unique identifier for each binary object. Ex. 1005, 4:35-47, 7:64-8:1.

We are not persuaded by PersonalWeb's argument that Woodhill does not use Binary Object Identifier 74, which is part of Binary Object Identification record 58, to access a particular binary object. Upon reviewing Woodhill's description of Binary Object Identification record 58, the only part of the record that uniquely identifies the binary object associated therewith is Binary Object Identifier 74. Ex. 1005, 4:45-47, 8:33-65. Therefore, during Woodhill's self-auditing procedure, we determine that Distributed Storage Manager program 24 uses Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25. *See* Ex. 1005, 18:16-19. Dr. Clark confirms such an operation was routine because it was old and well-known to access objects using their identifiers. *See* Ex. 1081 ¶ 20. We credit Dr. Clark's testimony because it is consistent with a general understanding of how one with ordinary skill in the art would use an identifier for basic file management functions, e.g., using an identifier to access a record stored in a database.

Next, we are not persuaded by PersonalWeb's argument that, during the self-auditing procedure, Binary Object Identifier 74 merely is used for comparison purposes after the particular binary object already has been accessed to determine if the audit restore worked properly. As we explained above, the only part of Binary Object Identification record 58 that uniquely identifies the binary object associated therewith is Binary Object Identifier 74. Ex. 1005, 4:45-47, 8:33-65. Consequently, during Woodhill's self-auditing procedure, Binary Object Identifier 74 serves the following two purposes: (1) Distributed Storage Manager program 24 uses Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25 (*see* Ex. 1005, 18:16-19); and (2) Binary Object Identifier 74, which is stored as part of the randomly selected Binary Object Identification record 58, is compared with Binary Object Identifier 74, previously calculated by Distributed Storage Manager program 24, in order to confirm whether the audit restore was successful (Ex. 1005, 18:28-38).

We also are not persuaded by PersonalWeb's argument that Woodhill's File Location 38 and File Name 40 in File Identification Record 34 are used to access a file containing a particular binary object, whereas Binary Object Stream Type 62 and Binary Object Offset 72 in Binary Object Identification record 58 are used to locate the binary object in that file. Although we recognize that a file containing a particular binary object may be accessed using File Location 38 and File Name 40 (Ex. 1005, 3:56-63), we nonetheless are persuaded that EMC has presented sufficient evidence to support a finding that a particular binary object or file also may be accessed using its Binary Object Identifier 74 (*see*, *e.g.*, Ex. 1005, 4:45-47, 8:33-65, 18:10-38).

Consequently, we agree with EMC that Woodhill's self-auditing procedure, which includes using Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25, describes the function of "accessing a particular data item using the identifier of the data item," as recited in dependent claim 4.

b. "access means"

Dependent claim 4 recites, in relevant part, "*access means* for accessing a particular data item using the identifier of the data item." Ex. 1001, 39:40-41 (emphasis added).

In its Patent Owner Response, PersonalWeb contends that Woodhill fails to disclose the corresponding structure identified for the claimed "access means." PO Resp. 29-30. In particular, PersonalWeb argues that, according to the specification of the '791 patent, the corresponding structure for this means-plus-function limitation includes looking to True File registry 126 for the record of a corresponding True Name. *Id*. at 29. PersonalWeb alleges that Woodhill fails to disclose such structure or anything equivalent thereto. *Id*.

In its Reply, EMC contends that PersonalWeb's argument is predicated on an improper claim construction for the claimed "access means." Reply 10. In particular, EMC argues that PersonalWeb attempts to add a plurality of True Names for a plurality of files in True File registry 126 to the corresponding structure for the claimed "access means," as well as limit this means-plus-function limitation to both the format of the records and number records to be checked. *Id*.

As explained in our Decision to Institute, we identified the corresponding structure for performing the claimed function of "accessing a particular data item using the identifier of the data item" to be a data processor programmed according to steps S292 and S294 illustrated in Figure 17(a). Dec. 25-26. With respect to steps S292 and S294 illustrated in Figure 17(a), the specification of the '791 patent discloses "look[ing] to the True File registry 126 for a True File entry record 140 for a corresponding

True Name (Step S292). . . . If there is already a True File ID for the entry (Step S294), this mechanism's task is complete." Ex. 1001, 17:10-23.

Although we agree with PersonalWeb that the claimed function of "accessing a particular data item using the identifier of the data item" encompasses looking to True File registry 126 for the record of a corresponding True Name, we nonetheless are persuaded that EMC has presented sufficient evidence to support a finding that Woodhill's Distributed Storage Manager program 24 may look to File Database 25 for Binary Object Identification record 58 of a corresponding Binary Object Identifier 74. As we explained previously, Woodhill's Distributed Storage Manager program 24 may access a particular binary object or file by using its Binary Object Identifier 74 to retrieve its corresponding Binary Object Identification record 58 in File Database 25. *See, e.g.*, Ex. 1005, 4:45-47, 8:33-65, 18:10-38.

Consequently, we agree with EMC that Woodhill's self-auditing procedure, which includes accessing a randomly selected binary object by using its Binary Object Identifier 74 to retrieve its corresponding Binary Object Identification record 58 in File Database 25, describes the corresponding structure for performing the function of "accessing a particular data item using the identifier of the data item," as recited in dependent claim 4.

c. "data associating means"

Dependent claim 4 recites, in relevant part, "*data associating means* for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item. Ex. 1001, 39:37-39 (emphasis added).

In its Petition, EMC contends that Woodhill discloses the "data associating means," as claimed. Pet. 58 (citing Ex. 1005, 7:60-8:65, 18:11-23). In particular, EMC argues that Woodhill's Distributed Storage Manager program 24, which executes on a computer, accesses and checks Binary Object Identification records 58 in File Database 25 to break up a plurality of files into one or more data streams, each of which is divided into one or more binary objects. Ex. 1041, 16 (citing Ex. 1005, fig. 5A). EMC also argues that Woodhill's Distributed Storage Manager program 24 executes on a computer to make and maintain Binary Object Identification record 58, which associates each binary object with its Binary Object Identifier 74. *Id*. EMC offers the testimony of Dr. Clark to support its position. Ex. 1009 ¶¶ 92, 93.

In its Patent Owner Response, PersonalWeb contends that the claimed function associated with the "data associating means" includes deleting a file in response to comparing True Names. PO Resp. 30. PersonalWeb argues that Woodhill fails to disclose this deletion function or anything equivalent thereto. *Id*. PersonalWeb further argues that Woodhill teaches away from this deletion function because Woodhill discloses that the purpose of the backup procedure is to ensure that backup copies of the binary objects are saved—not deleted or lost. *Id*. at 30-31.

In its Reply, EMC contends that the critical aspect of the claimed function associated with the "data association means" is to avoid unwanted duplicates. Reply 11. EMC argues that Woodhill performs this function because it detects and avoids unwanted duplicates in File Database 25. *Id*. (citing Ex. 1005, 9:23-27; Ex. 1081 ¶ 26). In particular, EMC argues that, during Woodhill's backup procedure, Woodhill prevents unwanted duplicates before they happen by determining which parts of a file have changed, and only backing up that changed data. *Id*. at 12 (Ex. 1005, 9:24-25; Ex. 1081 ¶ 26).

As explained in our Decision to Institute, we identified the corresponding structure for performing the claimed function of "making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item," to be a data processor programmed according to steps S230, S232, and S237-239 illustrated in Figure 11. Dec. 23-25. With respect to steps S230, S232, and S237-239 illustrated in Figure 11, the specification of the '791 patent discloses the following:

> First, determine the True Name of the data item corresponding to the given scratch File ID using the Calculate True Name primitive mechanism (Step S230). Next, look for an entry for the True Name in the True File registry 126 (Step S232) and determine whether a True Name entry, record 140, exists in the True File registry 126. If the entry record includes a corresponding True File ID or compressed File ID (Step S237), delete the file with the scratch File ID (Step S238). Otherwise store the give True File ID in the entry record (step S239).

Ex. 1001, 14:51-60.

Contrary to PersonalWeb's arguments, we construed the claimed function associated with the "data associating means" to encompass detecting and avoiding duplicate True File IDs in True File registry 126. *See, e.g.*, Ex. 1001, 14:51-60, fig. 11, S230, S232, S237-239. The claimed function is not limited to deleting a file in response to comparing a True File ID. *See, e.g.*, Ex. 1001, fig. 11, step S238. We agree with EMC that, during Woodhill's backup procedure, Distribute Storage Manager program 24 detects and avoids duplicate Binary Object Identifiers 74 in File Database 25 by determining which parts of a binary object or file have changed, and only backing up the changed data. Ex. 1005, 9:23-27. Dr. Clark also testifies that the claimed function associated with the "data associating means," and the operations performed during Woodhill's backup procedure, perform the identical function because they each assimilate data items without creating duplicates. Ex. 1081 ¶ 26. We credit Dr. Clark's testimony because it is consistent with our claim construction for the "data associating means," as well as Woodhill's description of the backup procedure.

To the extent PersonalWeb contends that Woodhill teaches away from the claimed function for the "data associating means" because the purpose of its backup procedure is to ensure that backup copies of the binary objects are saved—not deleted or lost—we disagree. PO Resp. 30. PersonalWeb's argument is not persuasive because EMC's proposed ground of unpatentability is based on anticipation by Woodhill. It is well settled that "[t]eaching away is irrelevant to anticipation." *Seachange Int'l, Inc., v. C-Cor, Inc.*, 413 F.3d 1361, 1380 (Fed. Cir. 2005).

In summary, we agree with EMC that Woodhill's backup procedure, which includes detecting and avoiding duplicate Binary Object Identifiers 74 in File Database 25 by determining which parts of a binary object or file have changed, and only backing up the changed data, describes the claimed function associated with the "data associating means," as recited in dependent claim 4. For the foregoing reasons, we conclude that that EMC has demonstrated by a preponderance of the evidence that dependent claim 4 is anticipated by Woodhill.

7. Claim 29

Dependent claim 29 recites "a data item is at least one of a file, a database record, a message, a data segment, a data block, a directory, and an instance [of] an object class." Ex. 1001, 42:54-57. The contentions and supporting evidence presented by EMC that explain how Woodhill describes the claimed subject matter recited in dependent claim 29 have merit and otherwise are unrebutted by PersonalWeb. Pet. 58 (citing Ex. 1005, 7:51-55); *see* Ex. 1009 ¶¶ 83-96. Therefore, we conclude that EMC has demonstrated by a preponderance of the evidence that dependent claim 29 is anticipated by Woodhill.

8. Claim 30

a. "accessing a data item in the system using the identifier of the data item"

Dependent claim 30 recites, in relevant part, "accessing a data item in the system using the identifier of the data item." Ex. 1001, 42:66-67. PersonalWeb relies upon essentially the same argument presented against dependent claim 4 to rebut the explanations provided by EMC as to how Woodhill describes the above-identified method step recited in independent claim 30. *Compare* PO Resp. 23-29 *with* PO Resp. 31-36. For the same reasons discussed above with respect to dependent claim 4, PersonalWeb's arguments are not persuasive. Therefore, we conclude that EMC has demonstrated by a preponderance of the evidence that independent claim 30 is anticipated by Woodhill.

9. Claims 31 and 32

Dependent claim 31 recites:

> making and maintaining, for a plurality of data items present in the system, an association between each of the data items and the identifier of each of the data items, wherein said accessing of a data item accesses a data item via the association.

Ex. 1001, 43:2-6. Dependent claim 32 recites "assimilating a new data item into the system, by determining the identifier of the new data item and associating the new data item with its identifier." Ex. 1001, 43:8-10. The contentions and supporting evidence presented by EMC that explain how Woodhill describes the claimed subject matter recited in dependent claims 31 and 32 have merit and otherwise are unrebutted by PersonalWeb. Pet. 58 (citing Ex. 1005, 7:60-8:65, 18:11-23); *see* Ex. 1009 ¶¶ 83-96. Therefore, we conclude that EMC has demonstrated by a preponderance of the evidence that dependent claims 31 and 32 are anticipated by Woodhill.

10. Claim 41

a. EMC does not switch between different unrelated embodiments when explaining how Woodhill describes the "accessing" method step

Dependent claim 41 recites, in relevant part, "*[t]he method of claim 30*, wherein *said accessing further comprises*: for a given data identifier and for a given current location and a remote location in the system." Ex. 1001, 45:8-10.

In its Petition, EMC contends that Woodhill discloses "for a given data identifier . . . to the current location," as recited in dependent claim 41. Pet. 58 (Ex. 1005, 9:5-23). EMC argues that, during Woodhill's backup procedure, the data processing system only backs up changed binary objects since the previous backup. Ex. 1041, 21. EMC further argues that

Woodhill's data processing system backs up binary objects from local computers 20 on remote backup file server 12. *Id*.

In its Patent Owner Response, PersonalWeb contends that EMC relies upon the self-auditing procedure disclosed in Woodhill to describe the "accessing" method step recited in independent claim 30, yet EMC relies upon the backup procedure disclosed in Woodhill to describe the additional features of the same "accessing" method step recited in dependent claim 41. PO Resp. 36-37 (Ex. 1005, 9:5-23, 18:10-38; Ex. 1009 ¶¶ 94-96). PersonalWeb argues that EMC cannot switch between different unrelated embodiments in Woodhill when explaining how Woodhill describes the "accessing" method step, as recited in independent claim 30, and further recited in dependent claim 41. *Id*. at 37-38.

In its Reply, EMC contends that it relied on only one embodiment in Woodhill to describe the "accessing" method step, as recited in independent claim 30, and further recited in dependent claim 41. Reply 12. In particular, EMC argues that Woodhill's Distributed Storage Manager program 24 is a single structure divided into several distinct functions that are illustrated in Figures 5A through 5L. *Id*. (citing Ex. 1004, 4:62-67). EMC further argues that, when Woodhill describes Distributed Storage Manager program 24, it indicates that "each distinct function operates in cooperation with the other functions to form a unitary computer program." *Id*. (quoting Ex. 1005, 4:67-5:2). We agree with EMC that it only relies upon one embodiment to describe the "accessing" method step, as recited in independent claim 30, and further recited in dependent claim 41.

When determining whether EMC relies on a single embodiment in Woodhill to describe the claimed "accessing" method step, our inquiry is "not constrained to proceed example-by-example when reviewing an allegedly anticipating prior art reference. Rather, [we] must, while looking at the reference as a whole, conclude whether or not that reference discloses all elements of the claimed invention arranged as in the claim." *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1369 n.5 (Fed. Cir. 2008).

The relevant disclosure in Woodhill is reproduced below in its entirety:

> For explanation purposes, the Distributed Storage Manager program 24 is divided into several functions which will be discussed in turn. Those of ordinary skill in the art will recognize, however, that each of the distinct functions operates in cooperation with the other functions to form a unitary program. Those of ordinary skill in the art will also recognize that the following discussion illustrates the operation of the Distributed Storage Manager program 24 on a single local computer 20, although it should be understood that the Distributed Storage Manager program 24 operates in the same fashion on each local computer 20 on the networked computer system 10.

Ex. 1005, 4:64-5:9.

Woodhill then proceeds to provide separate and distinct explanations as to how Distributed Storage Manager program 24 handles the operations of the backup procedure and the self-auditing procedure. Ex. 1005, 9:5-23, 18:10-38. Therefore, contrary to PersonalWeb's argument, Woodhill's backup procedure and self-auditing procedure are not mutually exclusive embodiments, but rather are distinct functions that operate with other functions to form one unitary computer program—namely Woodhill's Distributed Storage Manager program 24. Consequently, we are not persuaded that EMC switches between different unrelated embodiments when explaining how Woodhill describes the "accessing" method step, as recited in independent claim 30, and further recited in dependent claim 41.

b. Woodhill's back-up procedure discloses the claimed subject matter recited in claim 41

Dependent claim 41 recites, in relevant part:

> determining whether the data item corresponding to the given data identifier is present at the current location, and based on said determining, if said data item is not present at the current location, fetching the data item from a remote location in the system to the current location.

Ex. 1001, 45:11-16.

In its Patent Owner Response, PersonalWeb contends that Woodhill's backup procedure fails to disclose the above-identified features recited in dependent claim 41. PO Resp. 41-43. According to PersonalWeb, the ordinary and customary meaning of the claimed term "fetch" is "to go after and return." *Id*. at 41 (quoting THE AMERICAN HERITAGE DICTIONARY 486 (1975) (Ex. 2004)). Based on that dictionary definition, PersonalWeb argues that, during Woodhill's backup procedure, a new binary object is simply transmitted to remote backup filer server 12—not fetched. *Id*. In its Reply, EMC maintains that it properly relied on the functions performed by Woodhill's Distributed Storage Manager program 24, which it maintains is a single embodiment that incorporates both the operations of the backup procedure and the self-auditing procedure. Reply 13 (citing Ex. 1041, 18, 20-21).

According to Woodhill, both its system and method for managing storage space on network computer system 10 include selectively copying a binary object stored in one storage area to another storage area. Ex. 1005, Abstract, 2:4-6, 25-27. In our view, this general description of Woodhill's invention applies to its backup procedure. For instance, during Woodhill's backup procedure, Distributed Storage Manager program 24 backs up each binary object by storing a compressed copy of the binary object in the following two locations: (1) on disk drive 19 associated with local computer 20 somewhere other than local computer 20 where the binary object was stored originally; and (2) on remote backup file server 12. Ex. 1005, 9:31-38. Therefore, if a binary object ever was lost or destroyed at an entire site, e.g., disk drive 19 on local computer 20 or remote backup file server 12, Woodhill indicates that a copy of the binary object stored in another storage area may be copied to that site. *See* Ex. 1005, 9:39-45.

Even if we accept PersonalWeb's definition of "fetch" as "to go after and return with" (Ex. 2004), Woodhill's backup procedure still discloses fetching a binary object from a remote location if it is no longer present, e.g., lost or destroyed, at a current location, as required by dependent claim 41. For instance, if a failure occurs at disk drive 19 on local computer 20, Distributed Storage Manager program 24 may determine whether a binary object still is present at that location, i.e., the claimed "current location," by examining the binary objects and their corresponding Binary Object Identifiers 74 stored on disk drive 19. If the binary object and its corresponding Binary Object Identifier 74 have been lost, destroyed, or are no longer present at disk drive 19, Distributed Storage Manager program 24 could fetch a copy of the binary object using its Binary Object Identifier 74 from remote backup file server 12, i.e., the claimed "remote location," and return it to disk drive 19.

Alternatively, if a failure occurs at remote backup file server 12, Distributed Storage Manager program 24 may determine whether a binary object still is present at that location, i.e., the claimed "current location," by examining the binary objects and their corresponding Binary Object Identifiers 74 stored in File Database 25. If the binary object and its corresponding Binary Object Identifier 74 have been lost, destroyed, or are no longer present at remote backup file server 12, Distributed Storage Manager program 24 could fetch a copy of the binary object using its Binary Object Identifier 74 from disk drive 19 on local computer 20, i.e., the claimed "remote location," and return it to remote backup file server 12.

PersonalWeb also reiterates its argument that Woodhill's backup procedure does not "access a data item in the system using the identifier of the data item," as required by dependent claim 41 based on its dependency from independent claim 30. PO Resp. 42-43. For the same reasons discussed above with respect to dependent claim 4 and independent claim 30, PersonalWeb's argument is not persuasive.

c. Woodhill determines whether a particular data item is "not present" at a given location

In its Patent Owner Response, PersonalWeb presents a number of arguments that are predicated on the notion that Woodhill only is capable of analyzing information for a single binary object or file stored at a given location, and is incapable of analyzing the other files stored at that location. PO Resp. 43-49. PersonalWeb also alleges that both parties agree that Woodhill is incapable of determining whether a particular data item is "not present" at a given location. *Id.* at 42-43, 48-49 (citing Ex. 2008, 143, 145, 150-151). Based on those arguments, PersonalWeb asserts that Woodhill does not disclose "based on said determining, if said data item is not present at the current location, fetching the data item from a remote location in the system to the current location," as recited in dependent claim 41. *Id.* at 43, 49.

In its Reply, EMC contends that PersonalWeb mischaracterizes Dr. Clark's testimony by asserting that he agreed that it is impossible for Woodhill to determine whether a particular data item is not present at a given location. Reply 14. Instead, EMC argues that Dr. Clark only agreed that the hypothetical proposed by PersonalWeb during cross examination made this impossible—not that it was, in fact, impossible for Woodhill to determine whether a particular data item is not present at a given location. *Id.* EMC further contends that Woodhill is capable of determining whether the current version of a binary object or file is not present at remote backup server 12, i.e., the claimed "current location," and transmitting the file to that server. *Id.* (citing Ex. 1081 ¶ 11).

As we explained previously, Woodhill discloses that File Database 25 stores a plurality of Binary Object Identifiers 74 associated with different binary objects or files that have been backed up in the system. *See, e.g.*, Ex. 1005, 3:49-52, 4:30-34, 9:8-22. Dr. Clark testifies that, during Woodhill's backup procedure, Distributed Storage Manager program 24 determines whether a binary object or file is present at remote backup file server 12 by confirming that its Binary Object Identifier 74 already exists among the plurality of Binary Object Identifiers 74 stored in File Database 25. Ex. 1009 ¶ 89; *see* Ex. 1081 ¶ 14. Dr. Clark also testifies that Distributed Storage Manager program 24 only transmits the binary object or file to remote backup file server 12 if it is not already present at that location.

Ex. 1081 ¶ 11. We credit Dr. Clark's testimony because it is consistent with Woodhill's general disclosure of copying binary objects stored in one storage area to another storage area (Ex. 1005, 2:4-6, 25-27), as well as how Woodhill's Distributed Storage Manager program 24 transmits new or changed binary objects or files to remote backup file server 12 (Ex. 1005, 9:36-38).

We are not persuaded by PersonalWeb's allegation that the parties agree that it is "impossible" for Woodhill to determine whether a particular data item is "not present" at a given location. Dr. Clark stated that he did make such an admission and does not agree with PersonalWeb's assertion. Ex. 1081 ¶ 11. Upon reviewing the cited pages in the transcript of PersonalWeb's deposition of Dr. Clark, we note that the questions posed by PersonalWeb's counsel to Dr. Clark are couched in hypotheticals, and not directed to Woodhill's disclosure. *See, e.g.*, Ex. 2008, 143 ("Let's assume we have file A and File B. They're different files. Each of them has a plurality of binary objects. . . . Assume that the exact same binary object is actually present in both file A and file B."); Ex. 2008, 153 ("Assume you are given a sequence of bits, and you have a thousand files stored in a server, and you only have the capability of figuring out if that sequence of bits is in only one of those files, and you do not have the capability of figuring out if that sequence of bits is in the other 999 of those files."); Ex. 1081 ¶ 12. It is not clear to us how each of those constrained hypotheticals relates to the backup procedure disclosed in Woodhill, much less how a conclusion can be drawn from Dr. Clark's response to each hypothetical that he readily admitted it is "impossible" for Woodhill to determine whether a particular data item is "not present" a given location.

In summary, we agree with EMC that Woodhill's backup procedure, which includes determining whether a binary object or file corresponding to Binary Object Identifier 74 is present at remote backup file server 12 and, if not, transmitting it to that location, describes determining whether a particular data item is "not present" at a given location, as required by dependent claim 41. For the foregoing reasons, we conclude that EMC has demonstrated by a preponderance of the evidence that dependent claim 41 is anticipated by Woodhill.

11. Claim 33

a. Woodhill determines whether a particular data item is "not present" at a destination location

Independent claim 33 recites, in relevant part:

> determining, using the data identifier, whether the data item is present at the destination location; and based on the determining whether the data item is present, providing the destination location with the data item only if the data item is not present at the destination [location].

Ex. 1001, 43:19-23. PersonalWeb relies upon essentially the same arguments presented against dependent claim 41 to rebut the explanations provided by EMC as to how Woodhill describes the above-identified features recited in dependent claim 33. *Compare* PO Resp. 43-51 *with* PO Resp. 51-57. For the same reasons discussed above with respect to dependent claim 41, PersonalWeb's arguments are not persuasive.

b. Woodhill describes providing the destination location with a data item "only if" it is determined that the data item is not present at that destination location

PersonalWeb contends that, because Woodhill cannot determine whether a particular data item is "not present" at a destination location, Woodhill cannot disclose providing the destination location with a data item "only if" it is determined that the data item is not present at that destination location, as required by independent claim 33. PO Resp. 57 (citing Ex. 2013 ¶ 95). As we have explained previously, during Woodhill's backup procedure, Distributed Storage Manager program 24 determines whether a binary object or file corresponding to Binary Object Identifier 74 already exists on remote backup server 12 and, if not, transmits the binary object or file to that location. *See* Ex. 1005, 3:49-52, 4:30-34, 9:1-38; Ex 1009 ¶ 89; Ex. 1081 ¶¶ 11, 12, 14. In that scenario, Woodhill's remote backup file server 12 constitutes the claimed "destination location." For the foregoing reasons, we conclude that EMC has demonstrated by a preponderance of the evidence that independent claim 33 is anticipated by Woodhill.

D. Obviousness over Woodhill—Claims 1-4 and 29

EMC contends that claims 1-4 and 29 are unpatentable under § 103(a) over Woodhill. Pet. 59. In support of that alleged ground of unpatentability, EMC provides explanations as to how Woodhill teaches or suggests each claim limitation. *Id*. (citing Ex. 1041). EMC also submits declarations of Dr. Clark (Ex. 1009 ¶¶ 97-98; Ex. 1081) to support its positions. Upon reviewing EMC's Petition and supporting evidence, as well as PersonalWeb's Patent Owner Response and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claims 1-4 and 29 are obvious over Woodhill.

We begin our analysis with the principles of law that generally apply to a ground of unpatentability based on obviousness, and then we turn to the arguments presented by both EMC and PersonalWeb that are directed to whether Woodhill, as a whole, would have taught or suggested the "identity means" recited in independent claim 1 to one with ordinary skill in the art.

1. Principles of Law

A patent claim is unpatentable under § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc*., 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, which include the following: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). We analyze the ground of unpatentability based on obviousness over Woodhill with the above-identified principles in mind.

2. PersonalWeb's Contentions

a. There are no deficiencies in Woodhill to cure

At the outset, PersonalWeb contends that EMC's contentions regarding obviousness do not cure the deficiencies in Woodhill that are discussed above with respect to claims 1-4 and 29. PO Resp. 58-59. As we explained in our discussion of the ground of unpatentability based on anticipation by Woodhill, there are no such deficiencies in Woodhill to cure.

b. Woodhill, as a whole, would have taught or suggested the claimed "identity means" to one with ordinary skill in the art

Independent claim 1 recites, in relevant part:

> identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.

Ex. 1001, 39:16-21.

In its Petition, EMC contends that, to the extent that the claimed "identity means" requires an MD5 hash function, a person of ordinary skill in the art would have found it obvious to calculate Woodhill's Binary Object Identifier 74 for a particular binary object or file using an MD5 hash function. Pet. 59 (citing Ex. 1005, 8:52-58). According to Dr. Clark, this modification to Woodhill would constitute a simple substitution of one known element for another to obtain predictable results. *Id.* (citing Ex. 1009 ¶¶ 97, 98).

In its Patent Owner Response, PersonalWeb contends that it would not have been obvious to calculate Woodhill's Binary Object Identifier 74 using an MD5 has function because there were thousands, if not millions, of possible hash function known at the time of Woodhill's invention, and there would have been no logical reason to select an MD5 hash function for use in Woodhill. PO Resp. 59 (citing Ex. 2013 ¶ 98). PersonalWeb also argues that an MD5 hash function produces 16-byte hash values, whereas Woodhill desires a 4-byte hash value. *Id.* (citing Ex. 1005, 8:1-3).

In its Reply, EMC contends that PersonalWeb's argument only is relevant if we change our claim construction for "identity means" to include an MD5 hash function. Reply 15. In any event, EMC argues that Dr. Clark confirms that MD5 hash functions were old and well-known at the time of the invention of the '791 patent, and that use of an MD5 hash function in Woodhill's system would be a simple and obvious substitution. *Id.* (citing Ex. 1009 ¶¶ 97, 98; Ex. 1081 ¶¶ 27-29).

To the extent PersonalWeb now argues that the claimed "identity means" requires an MD5 hash function, we disagree. Similar to our explanation in the Decision to Institute, PersonalWeb's argument in that regard is not commensurate in scope with our claim construction of "identity means." *See* Dec. 27-28. The corresponding structure for performing the function of "determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier" is a data processor programmed to perform a hash function, e.g., MD5 or SHA. Neither the specification of the '791 patent, nor the claim itself, indicates that the "identity means" requires an MD5 hash function. Instead, an MD5 hash function is merely one of numerous hash functions capable of being programmed on a data processor that would satisfy this means-plus-function limitation.

Nonetheless, even if we assume that the claimed "identity means" requires an MD5 hash function, we agree with EMC that a person of ordinary skill in the art would have found it obvious to calculate Woodhill's Binary Object Identifier 74 for a particular binary object or file using an MD5 hash function. As discussed above, PersonalWeb asserts that an MD5 hash function produces 16-byte hash values. Woodhill discloses calculating Binary Object Identifier 74 for a binary object in various ways, including using a binary hash algorithm. Ex.1005, 8:1-31. Of importance here is that Woodhill discloses calculating Binary Object Hash field 70 against the content of the binary object taken one word or 16-bytes at a time. Ex. 1005, 8:23-24. Therefore, similar to PersonalWeb's assertion that an MD5 hash function produces 16-byte hash values, the binary hash algorithm disclosed in Woodhill also produces 16-byte hash values. Dr. Clark's testimony further confirms that Woodhill's Binary Object Identifier 74 and MD5 value are the same byte length, i.e., 16-bytes. Ex. 1081 ¶ 28.

In addition, we agree with PersonalWeb that one with ordinary skill in the art would have substituted an MD5 hash algorithm, which Dr. Clark confirms was old and well-known in the art at the time of the invention of the '791 patent (Ex. 1009 ¶ 97; Ex. 1081 ¶ 27), for Woodhill's binary hash algorithm. In our view, such a substitution is a predictable use of prior art elements according to their established functions—an obvious improvement. *See KSR*, 550 U.S. at 479.

In summary, PersonalWeb's assertion that the claimed "identity means" requires an MD5 hash function is not commensurate in scope with our claim construction for this mean-plus-function limitation. Nonetheless, even if we assume that the claimed "identity means" requires an MD5 hash function, EMC has presented sufficient evidence that Woodhill, as a whole, would have taught or suggested this means-plus-function limitation to one with ordinary skill in the art.

c. Secondary Considerations of Non-Obviousness—Licenses

In its Patent Owner Response, PersonalWeb contends that, because third parties have licensed the ’791 patent, evidence of non-obviousness exists that outweighs the evidence of obviousness based on Woodhill presented by EMC in this proceeding. PO Resp. 59-60. In support of its argument, PersonalWeb directs us to three licensing agreements (Exs. 2010-12), as well as the declaration of Kevin Bermeister (Ex. 2009 ¶¶ 3-9), and then argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*. at 60.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between claims 1-4 and 29 and the above-identified licensing agreements. Reply 15. EMC argues that each of the three licensing agreements granted rights to more than just claims 1-4 and 29, and involved related parties with interlocking ownership and business interests. *Id*. We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and the claimed subject matter recited in claims 1-4 and 29.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *GPAC,* 57 F.3d at 1580. Further, without a showing of nexus, “the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness” when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000); *see Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus in two respects. First, neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010-12) are directed to the claimed subject matter recited in claims 1-4 and 29. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in claims 1-4 and 29. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Second, even if we assume that the above-identified licenses establish some degree of industry respect for the claimed subject matter recited in claims 1-4 and 29, that success is outweighed by the strong evidence of obviousness over Woodhill discussed above.

Based on the record before us, including the evidence of obviousness based on Woodhill and the evidence of secondary considerations regarding licensing activities, we conclude that EMC has demonstrated by a preponderance of the evidence that claims 1-4 and 29 are obvious over Woodhill.

E. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following evidence: (1) paragraphs 13, 20, 24, 27, and 28 of the rebuttal declaration of Dr. Clark that rely on, and cite to, Peterson, Tanenbaum, Langer, and RFC 1321 because these paragraphs are irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding; (2) Langer, because it is not authenticated properly under Federal Rule of Evidence ("FRE") 901; (3) Langer, because it includes impermissible hearsay, in violation of FRE 802; (4) the "capable," "can," and "may" statements in Dr. Clark's rebuttal declaration because these statements are irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding; (5) a new contention that allegedly appears in Dr. Clark's rebuttal declaration because it is prejudicial, outside the scope of this proceeding, lacks foundation, lacks underlying facts and data, is in violation of FREs 702 and 705, and represents a new argument on reply; and (6) paragraphs 17-19 and 23 in Dr. Clark's rebuttal declaration because he relies upon subject matter in Woodhill that does not qualify as prior art to the '791 patent. Paper 62 ("PO Mot."). EMC opposes PersonalWeb's motion to exclude. Paper 71 ("Pet. Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 76 ("PO Reply"). For the reasons discussed below, PersonalWeb's motion to exclude is denied.

1. *The statements in Dr. Clark's rebuttal declaration regarding Peterson, Tanenbaum, Langer, and RFC 1321 are admissible evidence*

PersonalWeb contends that paragraphs 13, 20, 24, 27, and 28 of the rebuttal declaration of Dr. Clark (Ex. 1081) should be excluded because these paragraphs rely upon, and cite to, Peterson (Ex. 1075), Tanenbaum (Ex. 1076), Langer (Ex. 1003), and RFC 1321 (Ex. 1012). PO Mot. 1. PersonalWeb argues that this proceeding was only instituted based on Woodhill—not on Peterson, Tanenbaum, Langer, or RFC1321—and, therefore, EMC's reliance on these documents is outside the scope of this proceeding and impermissible. *Id*. In response, EMC contends that the statements regarding Peterson, Tanenbaum, Langer, and RFC 1321 are relevant to the instituted grounds of unpatentability based on Woodhill, and simply serve to corroborate the state of the art at the time of the '791 patent. Pet. Opp. 1. We agree with EMC.

The '791 patent has an effective filing date of April 11, 1995. Ex. 1001 at [63]. Peterson has a copyright date of 1983 (Ex. 1075, 2), Tanenbaum has a copyright date of 1987 (Ex. 1076, 2), Langer has publication date of August, 7, 1991 (Ex. 1003, 1), and RFC 1321 is dated April 1992 (Ex. 1012, 1). Each of these references has a publication date prior to April 11, 1995. We recognize that these prior art documents were relied on by EMC's rebuttal declarant, Dr. Clark, and are of the type that experts in the pertinent field reasonably would rely on to formulate their opinions. In other words, EMC may rely on these prior art documents to demonstrate what one with ordinary skill in the art would have known about technical features and developments in the pertinent art at the time of the '791 patent.

For the foregoing reasons, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude paragraphs 13, 20, 24, 27, and 28 of the rebuttal declaration of Dr. Clark.

2. EMC provides sufficient evidence to support a finding that Langer has been authenticated properly

PersonalWeb contends that EMC fails to provide evidence indicating that Langer (Ex. 1003) existed prior to the effective filing date of the '791 patent—April 11, 1995—and, therefore, should be excluded under FRE 901. PO Mot. 2-3. In particular, PersonalWeb argues that Langer allegedly was downloaded from the Internet in 2003 based on the "7/29/2003" date in the lower, right-hand corner. *Id*. at 2. PersonalWeb also argues that authentication of Langer requires personal knowledge of its existence prior to April 11, 1995. *Id*. at 3. In response, EMC contends that it submitted sworn testimony from Mr. Keith Moore that properly authenticates Langer under FREs 901(b)(1) and (4), 901(b)(3), 901(b)(8), and 901(b)(6) and (7). Pet. Opp. 2-3 (citing Ex. 1052 ¶¶ 5-11). In reply, PersonalWeb contends that Langer is not authenticated properly under the FREs identified by EMC. PO Reply. 1-5.

We agree with EMC that Langer has been authenticated properly under FRE 901(b)(1) and (4) because Mr. Moore testified that Langer is a periodical that was posted on Usenet newsgroups on August 7, 1991 (Ex. 1052 ¶¶ 11-15), and it includes distinct header fields unique to Usenet formatting and content (*id*. at ¶¶ 16,17). Although PersonalWeb presents several theories that attack the authenticity of Langer, PersonalWeb fails to explain adequately why the testimony offered by Mr. Moore does not authenticate Langer. PersonalWeb simply presents mere attorney arguments and does not offer testimony from its own expert contrary to the testimony offered by Mr. Moore. Therefore, based on the record before us, EMC has presented sufficient evidence to support a finding that Langer has been authenticated properly under FRE 901(b)(1) and (4).

We also are not persuaded by PersonalWeb's argument that the download date of "7/29/2003" in the lower, right-hand corner calls into question whether Langer existed prior to April 11, 1995. The mere fact that a "downloaded" copy of Langer has a date subsequent to the earliest effective filing date is not sufficient to rebut EMC's supporting evidence that

Langer is what it claims to be—namely a periodical posted on Usenet newsgroups on August 7, 1991. *See, e.g.*, Ex. 1052 ¶¶ 11-17.

To the extent PersonalWeb argues that Mr. Moore cannot authenticate Langer because he does not have personal knowledge of its existence prior to April 11, 1995, or that Mr. Albert Langer is the only person that can authenticate Langer properly, we disagree. Neither a declaration from Mr. Langer, nor evidence of someone actually viewing Langer prior to April 11, 1995, is required to support a finding that Langer is what it claims to be. *See In re Wyer*, 655 F.2d 221, 226 (CCPA 1981) (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication."); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would have known of the invention.).

For the foregoing reasons, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Langer as unauthenticated evidence.

3. Langer is not inadmissible hearsay

PersonalWeb contends that the dates in Langer, or any other information that purports to establish a publication date for Langer, are inadmissible hearsay under FRE 802 and not subject to any hearsay exception. PO Mot. 4. PersonalWeb also argues that, to the extent that EMC contends that any statements in Langer were made prior to the critical date of the '791 patent, the entirety of Langer is inadmissible hearsay. *Id*. In response, EMC contends that Langer is not hearsay because it is being offered for what it describes—not for the truth of its disclosure. Pet. Opp. 4. EMC also argues that the August 7, 1991, posting date on Langer's header and uniform resource locator ("URL") both were generated automatically by the hosting computer and, therefore, are admissible as non-hearsay to prove Langer's August 1991 publication date. *Id*. (citing Ex. 1052 ¶ 7). In reply, PersonalWeb maintains that the dates and other information in Langer used to establish its availability as of August 1991 amount to inadmissible hearsay. PO Reply 5.

We recognize that EMC's rebuttal declarant, Mr. Moore, reasonably would rely on the date of August 7, 1991, that appears in both Langer's header and URL to formulate his opinion on whether Langer was available publicly as of that date. Accordingly, the date of August 7, 1991, posted in Langer need not be admissible for the testimony of Mr. Moore to be admissible. Nonetheless, we agree with EMC that the date of August 7, 1991, posted on Langer's header and URL, serve a non-hearsay purpose for which it can be admitted—namely to prove that the document was available publicly as of that date.

Moreover, we are not persuaded by PersonalWeb's arguments that Langer, in its entirety, constitutes hearsay. With the exception of the dates in Langer, PersonalWeb does not identify specifically the textual portions of Langer that allegedly are being offered for the truth of the matter asserted, yet does seek to exclude Langer in its entirety. We will not go through the entirety of Langer and determine which portions PersonalWeb believes to be hearsay—this is something that PersonalWeb should have done in its motion to exclude.

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude the dates posted in Langer, or any statements made therein, as impermissible hearsay.

4. The "capable", "can," and "may" statements in Dr. Clark's rebuttal declaration are admissible

PersonalWeb contends that the "capable," "can," and "may" statements in Dr. Clark's rebuttal declaration (Ex. 1081) should be excluded because these statements are irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding. PO. Mot. 5-6 (citing FREs 401, 402, 403). In particular, PersonalWeb argues that the issue in this proceeding is what Woodhill discloses, or what is necessarily present in Woodhill, not what Woodhill is "capable" of or "may" do according to Dr. Clark. *Id*. at 5. In response, EMC contends that the "capable," "can," and "may" statements in Dr. Clark's rebuttal declaration were offered in response to arguments presented by PersonalWeb in its Patent Owner Response. Pet. Opp. 5-6 (citing PO Resp. 43-51; Ex. 2013 ¶¶ 46, 52). EMC argues that Dr. Clark was explaining what a person of ordinary skill in the art, upon reading Woodhill, would have understood Woodhill to disclose. *Id*. at 6.

We recognize that the focus of this proceeding is on the instituted grounds of unpatentability based on anticipation by, or obviousness over, Woodhill. Any statements that Dr. Clark makes regarding those grounds of unpatentability simply would affect how we weigh the testimony offered by Dr. Clark. When weighing the evidence provided by both parties, we are capable of determining whether Woodhill anticipates or renders obvious the challenged claims without being confused, misled, or prejudiced by Dr. Clark's testimony.

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude the "capable," "can," and "may" statements in Dr. Clark's rebuttal declaration.

5. *Dr. Clark's statements are direct rebuttal to an argument raised by PersonalWeb in its patent owner response*

PersonalWeb contends that Dr. Clark's rebuttal declaration includes a new contention not presented previously with the Petition that should be excluded because it is prejudicial, outside the scope of this proceeding, lacks foundation, lacks underlying facts and data, is in violation of FREs 702 and 705, and represents a new argument on reply. PO Mot. 5-6 (citing Ex. 1081 ¶ 19). In response, EMC contends that Dr. Clark's testimony properly responds to an argument presented by PersonalWeb in its Patent Owner Response. Pet. Opp. 6 (citing PO Resp. 21).

Based on our review of the argument presented by PersonalWeb in its Patent Owner Response, as well as the relevant portion of Dr. Clark's rebuttal declaration, we agree with EMC that Dr. Clark's testimony is direct rebuttal to PersonalWeb's argument that Woodhill does not enable comparing Binary Object Identifier 74 with one or more other Binary Object Identifiers 74. *Compare* PO Resp. 21 *with* Ex. 1081 ¶ 19. In other words, Dr. Clark's statement regarding enablement falls within the purview of 37 C.F.R. § 42.23(b), which provides that a petitioner's reply may only respond to arguments raised in the corresponding patent owner response.

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Dr. Clark's statements regarding enablement.

6. The statements in Dr. Clark's rebuttal declaration that rely on the claim language of Woodhill are admissible

PersonalWeb contends that paragraphs 17-19 and 23 of Dr. Clark's rebuttal declaration (Ex. 1081) that rely upon, and cite to, the claims of Woodhill should be excluded as irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding. PO Mot. 6. In particular, PersonalWeb argues that the "name" of a particular binary object identifier, as recited in the claims of Woodhill, is not prior art to the '791 patent because there is not sufficient written description support in Woodhill's original disclosure for that claimed subject matter. *Id.* at 6-7. In response, EMC contends that Woodhill's specification provides sufficient written description support for the "name" of a particular binary object identifier, as recited in the claims of Woodhill. Pet. Opp. 7 (Ex. 1005, 2:14-17, 7:60-8:65, 18:16-23, fig. 3).

Contrary to PersonalWeb's argument, Woodhill's original disclosure contains sufficient written description support for the "name" of a particular binary object identifier, as recited in the claims of Woodhill. Upon reviewing the description of Binary Object Identification record 58 in Woodhill's original disclosure, the only part of the record that uniquely identifies the binary object associated therewith is Binary Object Identifier 74. Ex. 2007, 26, 33-34. During Woodhill's self-auditing procedure, Distributed Storage Manager program 24 uses Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25. *See* Ex. 2007, 53. Dr. Clark confirms such an operation was routine because it was old and well-known to access records stored in a database using their identifiers. *See* Ex. 1081, ¶ 20.

Based on the cited portions in Woodhill's original disclosure, as well as Dr. Clark's corroborating testimony, we are persuaded that Woodhill's original disclosure conveys with reasonable clarity to one with ordinary skill in the art that Binary Object Identifier 74 may be considered a "name" for a binary object associated therewith because it uniquely identifies that binary object. *See Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc) (The written description test is whether the original disclosure of the application relied upon reasonably conveys to a person of ordinary skill in the art that the inventor had possession of the claimed subject matter as of the filing date.)

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude paragraphs 17-19 and 23 of Dr. Clark's rebuttal declaration that rely upon, and cite to, the "name" of a particular binary object identifier, as recited in the claims of Woodhill.

F. EMC's Motion to Exclude

EMC seeks to exclude three license agreements (Exs. 2010-12), as well as the two declarations offered by Mr. Kevin Bermeister (Exs. 2009, 2014) relating to those license agreements, because they are irrelevant under FRE 401, highly prejudicial, confusing, and misleading under FRE 403. Paper 65. PersonalWeb opposes EMC's motion to exclude. Paper 72. In response, EMC filed a reply to PersonalWeb's opposition to its motion to exclude. Paper 75.

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding the three license agreements (Exs. 2010-12) and the two declarations offered by Mr. Bermeister (Exs. 2009, 2014), we have concluded that EMC has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable. Accordingly, EMC's motion to exclude evidence is dismissed as moot.

III. CONCLUSION

EMC has demonstrated by a preponderance of the evidence that claims 1-4, 29-33, and 41 of the '791 patent are unpatentable based on the grounds of unpatentability set forth in the table below.

Claims	**Basis**	**Reference**
1-4, 29-33, and 41	§ 102(e)	Woodhill
1-4 and 29	§ 103(a)	Woodhill

IV. ORDER

In consideration of the foregoing, it is

ORDERED that, based on a preponderance of the evidence, claims 1-4, 29-33, and 41 of the '791 patent are unpatentable;

FURTHER ORDERED that PersonalWeb's motion to exclude evidence is DENIED;

FURTHER ORDERED that EMC's motion to exclude evidence is DISMISSED as moot; and

FURTHER ORDERED that, because this is a final written decision, parties to this proceeding seeking judicial review of our decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

For PETITIONERS:

Peter Dichiara
David L. Cavanaugh
WILMER CUTLER PICKERING HALE AND DORR LLP
Peter.Dichiara@wilmerhale.com
David.Cavanaugh@wilmerhale.com

For PATENT OWNERS:

Joseph A. Rhoa
Updeep S. Gill
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

Trials@uspto.gov
571-272-7822

Paper 80
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION and VMWARE, INC.,
Petitioners,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00083
Patent 6,415,280 B1

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

ZECHER, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. BACKGROUND

EMC Corporation and VMware, Inc. (collectively, "EMC") filed a Petition on December 15, 2012, requesting an *inter partes* review of independent claims 36 and 38 of U.S. Patent No. 6,415,280 B1 (Ex. 1001, "the '280 patent"). Paper 6 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") timely filed a Patent Owner's Preliminary Response. Paper 14 ("Prelim. Resp."). Taking into account PersonalWeb's Preliminary Response, the Board determined that the information presented in the Petition demonstrated that there was a reasonable likelihood that EMC would prevail in challenging independent claims 36 and 38 as unpatentable under 35 U.S.C. §§ 102(e) and 103(a). Pursuant to 35 U.S.C. § 314, the Board instituted this proceeding on May 17, 2013, as to the challenged claims of the '280 patent. Paper 19 ("Dec.").

During this proceeding, PersonalWeb timely filed a Patent Owner Response (Paper 45, "PO Resp."), and EMC timely filed a reply to the Patent Owner Response (Paper 51, "Reply"). A consolidated oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This decision is a final written decision under 35 U.S.C. § 318(a) as to the patentability of the challenged claims. Based on the record before us, EMC has demonstrated

[1] This proceeding, as well as IPR2013-00082, IPR2013-00084, IPR2013-00085, IPR2013-00086, and IPR2013-00087, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 79.

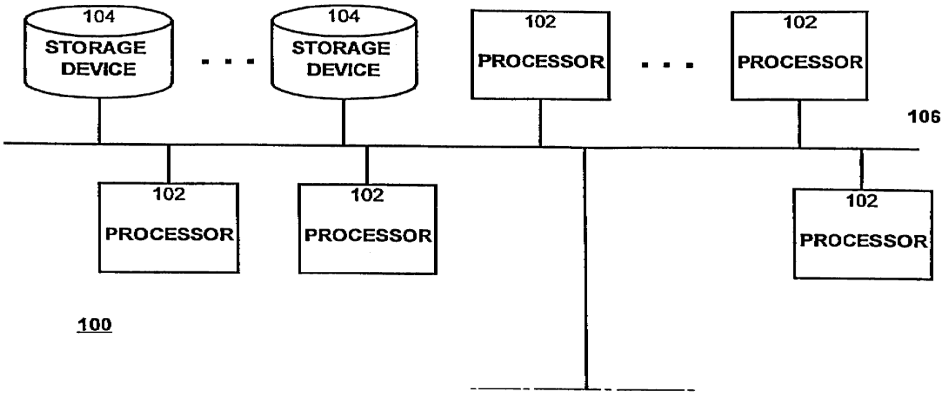
by a preponderance of the evidence that independent claims 36 and 38 are unpatentable.

A. The Invention of the '280 Patent

The invention of the '280 patent relates to a data processing system that identifies data items using substantially unique identifiers, otherwise referred to as True Names, which depend on all the data in the data item and only on the data in the data item. Ex. 1001, 1:12-16, 3:28-31, 6:7-9. According to the '280 patent, the identity of a data item depends only on the data and is independent of the data item's name, origin, location, address, or other information not directly derivable from the data. Ex. 1001, 3:32-34. The invention of the '280 patent also examines the identities of a plurality of data items in order to determine whether a particular data item is present in the data processing system. Ex. 1001, 3:35-38.

Figures 1(a) and 1(b) illustrate the data processing system that implements the invention of the '280 patent. Ex. 1001, 4:45-47. Figure 1(a) is reproduced below.

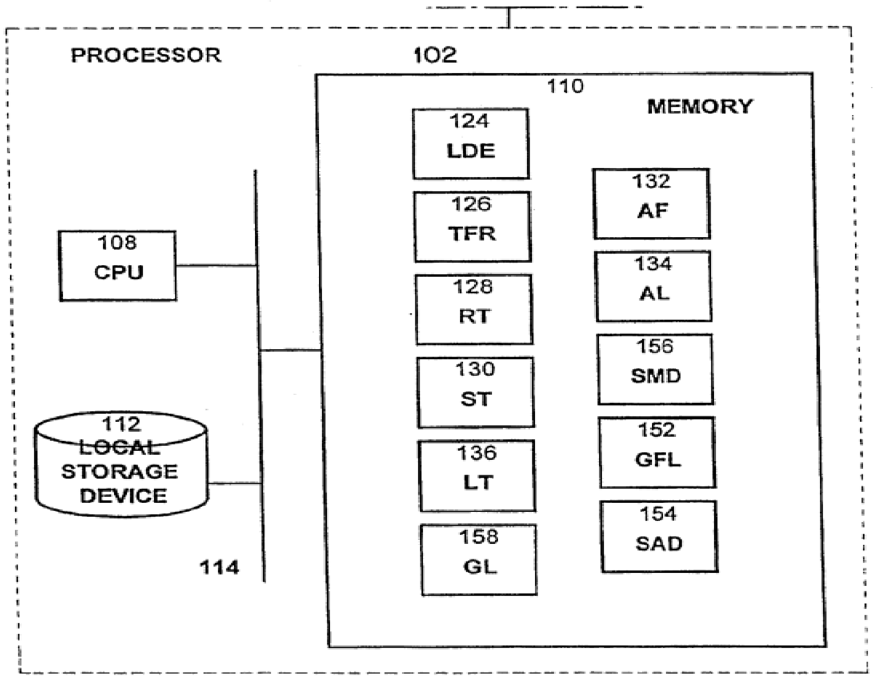
FIG. 1(a)



As shown in Figure 1(a), data processing system 100 includes one or more processors 102 and various storage devices 104 connected via bus 106. Ex. 1001, 4:59-64.

Figure 1(b) is reproduced below.

FIG. 1(b)



As shown in Figure 1(b), each processor 102 includes central processing unit 108, memory 110, and one or more local storage devices 112 connected via internal bus 114. Ex. 1001, 4:65-5:1. Memory 110 in each processor 102 stores data structures that are either local to the processor, itself, or shared amongst multiple processors in the data processing system. Ex. 1001, 7:65-8:13.

The ’280 patent further discloses accessing data items by referencing their identities or True Names independent of their present location in the data processing system. Ex. 1001, 34:20-22. The actual data item or True File corresponding to a given data identifier or True Name is capable of residing anywhere on the data processing system, i.e., locally, remotely, offline, etc. Ex. 1001, 34:22-24. If a requested data item or True File is local with respect to the data processing system, a prospective user can access the data in the True File. Ex. 1001, 34:24-26. If a requested data item or True File is not local with respect to the data processing system, a prospective user may use the True File registry to determine the location of copies of the True File according to its given True Name. Ex. 1001, 34:26-30. However, if for some reason a prospective user cannot locate a copy of the requested data item or True File, the processor employed by the user may invoke the Request True File remote mechanism to submit a general request for the data item or True File to all the processors in the data processing system. Ex. 1001, 34:34-40.

B. Challenged Claims

Independent claims 36 and 38 are the only claims challenged by EMC in this proceeding and are reproduced below:

> 36. A method of delivering a data file in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:
>
> storing the data file is [sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server*;* and
>
> responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client.

Ex. 1001, 43:54-63.

> 38. A method of delivering a data file in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

> storing the data file is [sic] on a first server and storing copies of the data file on a set of servers distinct from the first server; and
>
> responsive to a client request for the data file, the request including a value determined as a given function of the contents of the data file, providing the data file to the client.

Ex. 1001, 44:3-13.

C. Related Proceedings

EMC indicates that the ’280 patent was asserted against it in *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED, pending in the United States District Court for the Eastern District of Texas. Pet. 1. EMC also filed five other petitions seeking *inter partes* review of the following patents: (1) U.S. Patent No. 5,978,791 (“the ’791 patent”) (*EMC Corp. and VMware, Inc. v. PersonalWeb Techs., LLC*, IPR2013-00082); (2) U.S. Patent No. 7,945,544 (*EMC Corp. v. PersonalWeb Techs., LLC*, IPR2013-00084); (3) U.S. Patent No. 7,945,539 (*EMC Corp. v. PersonalWeb Techs., LLC*, IPR2013-00085); (4) U.S. Patent No. 7,949,662 (*EMC Corp. v. PersonalWeb Techs.*, LLC, IPR2013-00086); and (5) U.S. Patent No. 8,001,096 (*EMC Corp. v. PersonalWeb Techs., LLC*, IPR2013-00087). *Id.*

D. Prior Art Relied Upon

EMC relies upon the following prior art reference:

Woodhill	US 5,649,196	July 15, 1997 (effectively filed July 1, 1993)	Ex. 1005

E. Grounds of Unpatentability

We instituted this proceeding based on the grounds of unpatentability set forth in the table below.

Claims	Basis	Reference
36 and 38	§ 102(e)	Woodhill
36 and 38	§ 103(a)	Woodhill

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, we construe a claim by applying the broadest reasonable interpretation in light of the specification of the patent in which it appears. 37 C.F.R. § 42.100(b); *see* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). There is a "heavy presumption" that a claim term carries its ordinary and customary meaning. *CCS Fitness, Inc. v. Brunswick Corp*., 288 F.3d 1359, 1366 (Fed. Cir. 2002). However, a "claim term will not receive its ordinary meaning if the patentee acted as his own lexicographer and clearly set forth a definition of the disputed claim term in either the specification or prosecution history." *Id*. "Although an inventor is indeed free to define the specific terms used to describe his or her invention, this must be done with reasonable clarity, deliberateness, and precision." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

In its Petition, EMC identified six claim terms and provided a claim construction for those terms. Pet. 6-7. Those claim terms are listed as follows: (1) "data" and "data item"; (2) "file system"; (3) "file"; (4) "location"; (5) "local"; and (6) "True Name, data identity, and data identifier." *Id*. In the Decision to Institute, we indicated that only the claim terms "data" and "file" are used together as "data file" in independent claims 36 and 38. Dec. 10. Based on an explicit or special definition for the claim term "file" in the specification of the '280 patent, we construed the claim term "data file" as "a named data item, such as a simple file that includes a single, fixed sequence of data bytes or a compound file that includes multiple, fixed sequences of data bytes." *Id*. at 10-11 (citing Ex. 1001, 5:47-54). We also concluded that the preambles of independent claims 36 and 38 are entitled to patentable weight. *Id*. at 9-10. In its Patent Owner Response, PersonalWeb indicated that it agrees with our claim construction of the claim term "data file," as well as our conclusion that the preambles of independent claims 36 and 38 are entitled to patentable weight. PO Resp. 1-2 (quoting Dec. 10-11).

B. The Level of Ordinary Skill in the Art

In determining the level of one with ordinary skill in the art, we note that various factors may be considered, including "type of problems encountered in the art; prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field." *In re GPAC*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (citing *Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986)). There is sufficient evidence in the record before us that reflects the knowledge level of a person with ordinary skill in the art. PersonalWeb's expert, Dr. Robert B.K. Dewar, attests that a person with ordinary skill in the art would be an individual with a bachelor's degree in computer science who possesses ten to fifteen years of teaching or work experience in the field of data processing systems. Ex. 2013 ¶ 18.

C. Anticipation by Woodhill—Independent Claims 36 and 38

EMC contends that independent claims 36 and 38 are anticipated under § 102(e) by Woodhill. Pet. 39-47. In support of that alleged ground of unpatentability, EMC provides explanations as to how Woodhill describes each claim limitation. *Id*. (citing Ex. 1032). EMC also submits the declarations of Dr. Douglas W. Clark (Ex. 1009 ¶¶ 23-27; Ex. 1078) to support its positions. Upon reviewing EMC's Petition and supporting evidence, as well as PersonalWeb's Patent Owner Response and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that independent claims 36 and 38 are anticipated by Woodhill.

We begin our analysis with the principles of law that generally apply to a ground of unpatentability based on anticipation, followed by a brief discussion of Woodhill, and then we turn to the arguments presented by both EMC and PersonalWeb that are directed towards each challenged claim.

1. Principles of Law

To establish anticipation under § 102(e), "all of the elements and limitations of the claim must be shown in a single prior reference, arranged as in the claim." *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001). "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987). We analyze the ground of unpatentability based on anticipation by Woodhill with the above-stated principles in mind.

2. *Woodhill*

Woodhill generally relates to a system and method for distributed storage management on a networked computer system that includes a remote backup file server in communication with one or more local area networks. Ex. 1005, 1:11-17. Figure 1 of Woodhill, which is reproduced below, illustrates networked computer system 10. Ex. 1005, 2:56-58.

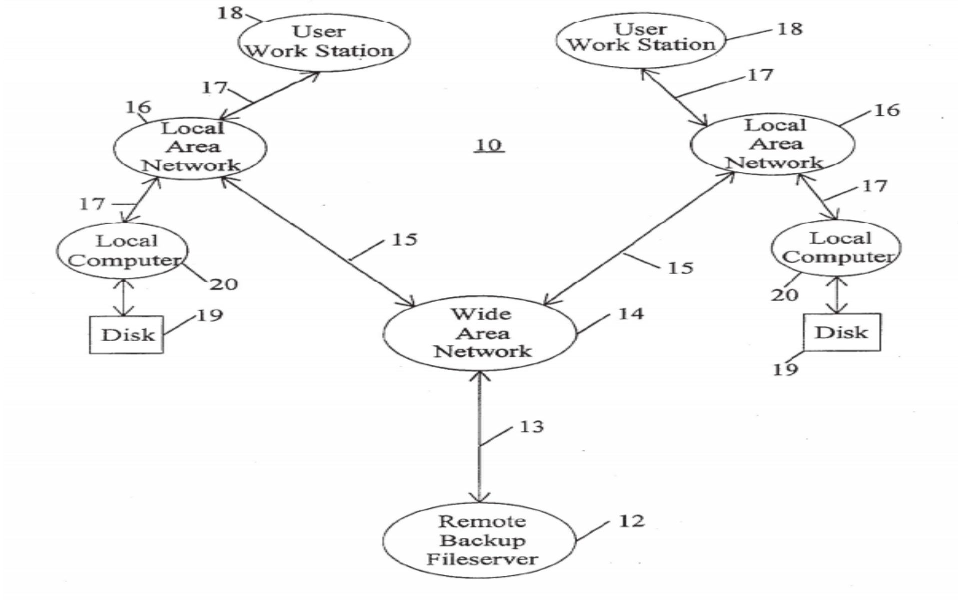


FIG. 1

As shown in Figure 1 of Woodhill, remote backup file server 12 communicates with wide area network 14 via data path 13, wide area network 14 communicates with a plurality of local area networks 16 via data paths 15, and each local area network 16 communicates with multiple user workstations 18 and local computers 20 via data paths 17. Ex. 1005, 3:12-31. The storage space on each disk drive 19 on each local computer 20 is allocated according to the hierarchy illustrated in Figure 2. Ex. 1005, 3:31-44.

Figure 2 of Woodhill, which is reproduced below, illustrates Distributed Storage Manager program 24 that allocates storage space on each of the storage devices in networked computer system 10. Ex. 1005, 2:59-62.

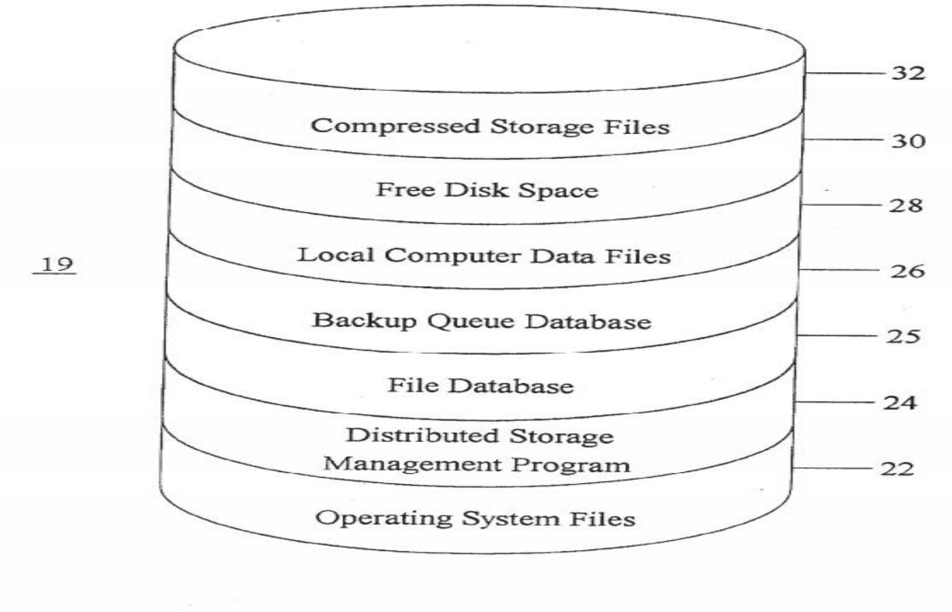


FIG. 2

As shown in Figure 2 of Woodhill, Distributed Storage Manager program 24 builds and maintains File Database 25 on the one or more disk drives 19 on each local computer 20 in networked computer system 10. Ex. 1005, 3:45-49. Distributed Storage Manager program 24 views a file as a collection of data streams. Ex. 1005, 4:13-15. Woodhill defines a data stream as a distinct collection of data within a file that may change independently from other distinct collections of data within the file. Ex. 1005, 4:15-18. For instance, Woodhill discloses that a file may contain both its normal data and any extended attribute data. Ex. 1004, 4:18-19. Depending on the size of the data stream, Distributed Storage Manager program 24 divides each data stream into one or more binary objects. Ex. 1005, 4:21-30.

Figure 3 of Woodhill, which is reproduced below, illustrates File Database 25 used by Distributed Storage Manager program 24. Ex. 1005, 2:63-64.

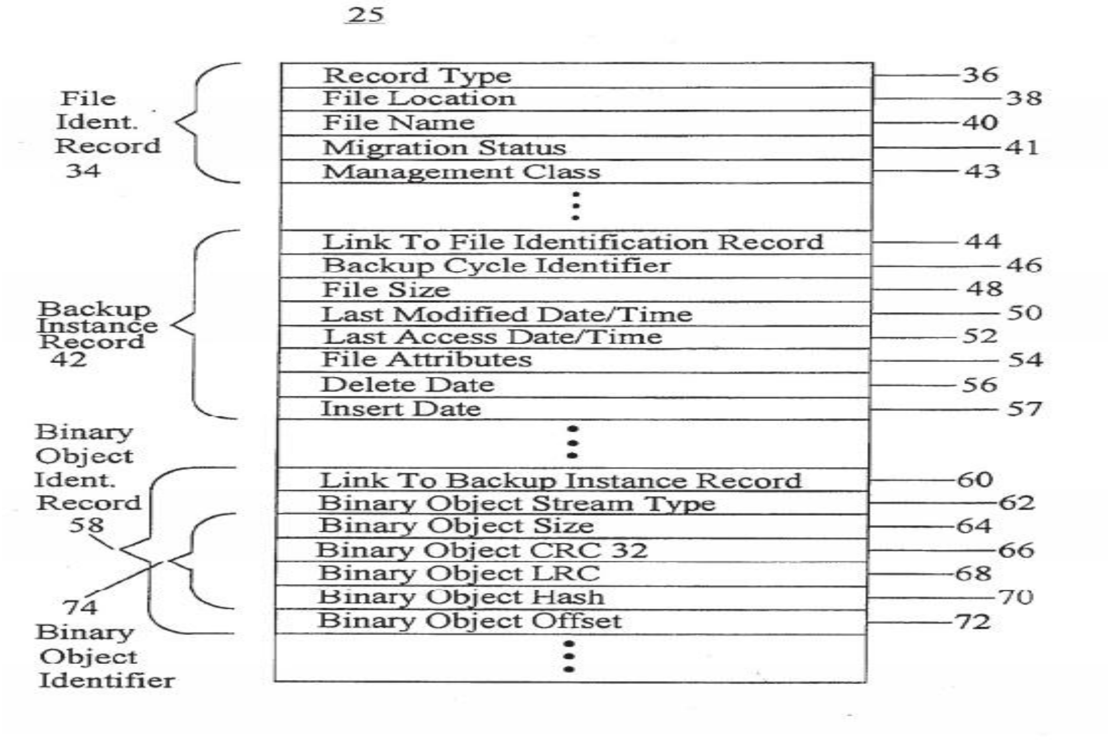


FIG. 3

As shown in Figure 3 of Woodhill, File Database 25 includes the following three levels of records organized according to a predefined hierarchy: (1) File Identification Record 34; (2) Backup Instance Record 42; and (3) Binary Object Identification Record 58. Ex. 1005, 3:54-4:47. Binary Object Identification Record 58 includes, amongst other things, Binary Object Identifier 74 that comprises Binary Object Size 64, Binary Object CRC32 66, Binary Object LRC 68, and Binary Object Hash 70.

Ex. 1005, 4:45-47, 7:64-8:1. Binary Object Identifier 74 is a unique identifier for each binary object that is backed up. Ex. 1005, 4:45-47.

Although Woodhill discloses calculating Binary Object Identifier 74 in various ways, e.g., using a binary hash algorithm (Ex.1005, 8:1-31), the key notion is that Binary Object Identifier 74 is calculated from the content of the data instead of from an external or arbitrary source. Ex. 1005, 8:38-42. In other words, Woodhill recognizes that the critical feature in creating Binary Object Identifier 74 is that the identifier should be based on the contents of the binary object, such that Binary Object Identifier 74 changes when the contents of the binary object changes. Ex. 1005, 8:58-62. Therefore, duplicate binary objects, even if resident on different types of computers in the network, may be recognized by their identical Binary Object Identifiers 74. Ex. 1005, 8:62-65.

Woodhill discloses that Distributed Storage Manager program 24 performs two backup operations concurrently. Ex. 1005, 9:30-31. First, Distributed Storage Manager program 24 stores a compressed copy of each binary object that it needs to restore disk drive 19 on each local computer 20 somewhere on local area network 16 other than on local computer 20 where the binary object originally resided. Ex. 1005, 9:31-36. Second, Distributed Storage Manager program 24 transmits new or changed binary objects to remote backup file server 12. Ex. 1005, 9:36-38.

Woodhill also discloses that Distributed Storage Manager program 24 performs auditing and reporting functions on a periodic basis to ensure that binary objects, which already have been backed up, may be restored. Ex. 1005, 18:11-13. Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by a Binary Object Identification Record 58 stored in File Database 25. Ex. 1005, 18:16-19.

3. Independent Claim 36

a. "a request for the data file from a client, where the request includes a hash of contents of the data file"

Independent claim 36 recites, in relevant part, "responsive to *a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client*." Ex. 1001, 43:62-64 (emphasis added).

In its Petition, EMC contends that Woodhill discloses that a local computer, i.e., a client, can request that a binary object be restored. Pet. 45 (citing Ex. 1005, 10:27-32). According to EMC, Dr. Clark confirms that such a request includes Binary Object Identifier 74 with a hash of the contents of the requested binary object. *Id*. (citing Ex. 1009 ¶ 26). EMC argues that the binary object is provided to the local computer in response to such a request. *Id*. In addition to relying upon Woodhill's backup procedure to support its position, EMC's claim chart also directs us to Woodhill's self-auditing procedure to describe the disputed limitation. Ex. 1032, 4-5 (citing Ex. 1005, 18:11-23).

In its Patent Owner Response, PersonalWeb contends that, during Woodhill's self-auditing procedure, a randomly selected binary object is identified by Binary Object Identification record 58 stored in File Database 25. PO Resp. 3-4 (citing Ex. 1005, 18:10-38; Ex. 2013 ¶¶ 102-04). PersonalWeb argues that Woodhill's Binary Object Identifier 74 is not included in any such "request," but rather is used for comparison purposes after the binary object associated therewith already has been accessed in order to determine if the audit restore worked properly. *Id*. at 4 (citing Ex. 1005, 18:28-38; Ex. 2013 ¶ 103). PersonalWeb further argues that, although Woodhill's Binary Object Identification record 58 identifies a particular binary object, Woodhill discloses that the record is stored in File Database 25 and never discloses that the record is part of the request for the binary object. *Id*. (citing Ex. 1005, 8:16-19; Ex. 2013 ¶ 104). PersonalWeb also asserts that EMC's expert, Dr. Clark, acknowledged that Woodhill fails to disclose that Binary Object Identifier 74 is part of a request. *Id*. at 5-6 (citing Ex. 2008, 167-68, 172-73).

In its Reply, EMC contends that, during Woodhill's self-auditing procedure, Distributed Storage Manager program 24 "initiates a restore of a . . . binary object identified by a Binary Object Identification Record 58." Reply 2 (quoting Ex. 1005, 18:12-20 (emphasis omitted)). EMC argues that Woodhill's Binary Object Identification Record 58 includes Binary Object Identifier 74, and the identifier, itself, includes Binary Object hash field 70 that represents a hash of the contents of the binary object. *Id*. (citing PO Resp. 4; *see also* Ex. 1005, 8:38-65, fig. 3; Ex. 1074, 136). According to EMC, Dr. Clark confirms that Binary Object Identifier 74 within Binary Object Identification Record 58 is used to identify and request binary objects to restore to the local computer. *Id*. at 3 (Ex. 1078 ¶¶ 8-15).

EMC further contends that, contrary to PersonalWeb's argument that Dr. Clark acknowledges that Woodhill fails to disclose that Binary Object Identifier 74 is part of the request, Dr. Clark has maintained unequivocally that the restore requests include Binary Object Identification Record 58 and that record clearly includes hashes in Binary Object Identifier 74. *Id*. at 4 (citing Ex. 2008, 216-17). EMC argues that it is self-evident that Woodhill's

Binary Object Identifier 74 within Binary Object Identification Record 58 is used to identify and access a binary object. *Id*. at 4-5. According to EMC, this is why Woodhill refers to Binary Object Identifier 74 as an "identifier," and also why independent claim 1 of Woodhill refers to it as a "name." *Id*. at 5 (citing Ex. 1005, 22:3-4).

As we explained in the Decision to Institute (Dec. 15-17), Woodhill discloses that Distributed Storage Manager program 24 performs auditing and reporting functions on a periodic basis in order to ensure that the binary objects, which already have been backed up, may be restored. Ex. 1005, 18:11-13. According to Woodhill, Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by Binary Object Identification Record 58 stored in File Database 25. Ex. 1005, 18:16-19. Binary Object Identification Record 58 includes, amongst other things, a Binary Object Identifier 74, which is a unique identifier for each binary object. Ex. 1005, 4:35-47, 7:64-8:1. Binary Object Identifier 74 includes, amongst other things, Binary Object Hash field 70, which is calculated against the contents of the binary object taken one word, i.e., 16-bytes, at a time using a binary hash algorithm. Ex. 1005, 7:64-8:32.

We are not persuaded by PersonalWeb's argument that Woodhill does not use Binary Object Identifier 74, which is part of Binary Object Identification record 58, to identify and request a particular binary object. Upon reviewing Woodhill's description of Binary Object Identification record 58, the only part of the record that identifies uniquely the binary object associated therewith is Binary Object Identifier 74. Ex. 1005, 4:45-47, 8:33-65. Moreover, Woodhill discloses that Binary Object Hash field

70, which is one of four fields that comprise Binary Object Identifier 74, is a hash of contents of the binary object, itself. Ex. 1005, 8:21-23.

Therefore, during Woodhill's self-auditing procedure, we determine that Distributed Storage Manager program 24 uses Binary Object Identifier 74 to identify and request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25. *See* Ex. 1005, 18:16-19. Given that Woodhill's Binary Object Identifier 74 includes Binary Object Hash field 70, such a request necessarily encompasses a hash of contents of the binary object, itself. Ex. 1005, 7:64-8:32. Dr. Clark confirms that such an operation was routine because it was old and well-known to identify and request objects using their identifiers. *See* Ex. 1078 ¶¶ 10, 11. We credit Dr. Clark's testimony because it is consistent with a general understanding of how one with ordinary skill in the art would use an identifier for basic file management functions, e.g., using an identifier to identify and request a record stored in a database.

Next, we are not persuaded by PersonalWeb's argument that, during the self-auditing procedure, Binary Object Identifier 74 merely is used for comparison purposes after the particular binary object already has been accessed to determine if the audit restore worked properly. As we explained above, the only part of Binary Object Identification record 58 that identifies uniquely the binary object associated therewith is Binary Object Identifier 74. Ex. 1005, 4:45-47, 8:33-65. Consequently, during Woodhill's self-auditing procedure, Binary Object Identifier 74 serves the following two purposes: (1) Distributed Storage Manager program 24 uses Binary Object Identifier 74 to request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25 (*see* Ex. 1005, 18:16-19); and (2) Binary Object Identifier 74, which is stored as part of the randomly selected Binary Object Identification record 58, is compared with Binary Object Identifier 74 previously calculated by Distributed Storage Manager program 24 in order to confirm whether the audit restore was successful (Ex. 1005, 18:28-38).

In summary, we agree with EMC that Woodhill's self-auditing procedure, which includes using Binary Object Hash field 70 in Binary Object Identifier 74 to identify and request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25, describes the method step of "responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client," as recited in independent claim 36.

b. "a data file" and "a hash of the contents of the data file"

Independent claim 36 recites, in relevant part, "a method of delivering *a data file* in a network. . . [and a] request including *a hash of the contents of the data file*." Ex. 1001, 43:55-63 (emphasis added).

In its Petition, EMC contends that Woodhill's binary objects constitute the claimed "data files." Pet. 44; *see* Ex. 1032, 3. In particular, EMC argues that Woodhill discloses storing binary objects, i.e., the claimed "data files," on local computer 20, and storing copies of the binary objects on at least one other local computer 20, as well as on remote backup file server 12. *Id*. at 44-45 (citing Ex. 1005, 4:14-26, 9:42-44, fig. 1); *see* Ex. 1032, 3 (citing Ex. 1005, 9:30-38).

In its Patent Owner Response, PersonalWeb contends that Woodhill's binary object is not a claimed "data file" because the binary object is not a "named data item." PO Resp. 6 (citing Ex. 2013 ¶¶ 105-06). In particular, PersonalWeb argues that Woodhill's binary objects are identified by respective Binary Object Identifiers 74; however, PersonalWeb asserts that Binary Object Identifiers 74 are not file names. *Id*. at 7 (citing Ex. 1005, 4:45-46; Ex. 2013 ¶ 106). PersonalWeb then argues that, because Woodhill's Binary Object Identifier 74 is not a file name, it follows that a binary object is not a "named data item." *Id*.

In addition, PersonalWeb argues that, according to the specification of the '280 patent, the claimed "a hash of the contents of the data file" requires that the hash must be *all the data* in the data file. *Id*. at 7-8 (citing Ex. 1001, 1:14-16, 3:29-31, 33:1-7) (emphasis added). PersonalWeb then argues that, although Woodhill discloses that Binary Object Identifier 74 includes Binary Object hash field 70 resulting from the application of a hash function to a binary object, Woodhill fails to disclose a hash of all the data in the data file, as required by independent claim 36. *Id*. at 8 (citing Ex. 2013 ¶ 106).

In its Reply, EMC contends that, contrary to PersonalWeb's arguments that Woodhill's binary objects are not named and Binary Object Identifiers 74 are not file names, independent claim 36 does not require that the request for a data file include a file name, or even a name. Reply 6. Instead, EMC argues that independent claim 36 only requires that the request be "for a data file," i.e., for a name data item, and that the request include "a hash of the contents of the data file." *Id*. at 6-7 (citing Ex. 1001, 43:62-64). EMC then relies upon its previous explanation of Woodhill's self-auditing procedure to support its position that Woodhill describes "a data file" and "a hash of the contents of the data file," as recited in independent claim 36. *Id*. at 7 (citing Ex. 1005, 17:18-45, 18:16-19; Ex. 1078 ¶¶ 8-15).

EMC further argues that, even if we were to accept PersonalWeb's overly narrow claim construction, Woodhill discloses data files that have only a single binary object, and Binary Object Identifier 74 is the name of the binary object associated therewith. Reply 8 (citing Ex. 1005, 1:66-2:3, 22:3-4). According to EMC, Dr. Clark confirms that Woodhill's data files that contain a single binary object may be named by their file name, in addition to being named by Binary Object Identifier 74. *Id*. at 9 (citing Ex. 1078 ¶¶ 4-7). EMC also disagrees with PersonalWeb's argument that the hash included in the request must be based on all the data in the data file because such a construction is overly narrow, is not consistent with the broadest reasonable interpretation of independent claim 36, and would exclude the preferred embodiment in the '280 patent. *Id*. at 7-8 (citing Ex. 1001, 21:30-50, 34:4-8; Ex. 1078 ¶ 14).

As we explained in the Decision on Institution, we construed the claim term "data file" as "a name data item, such as a simple file that includes a single, fixed sequence of data bytes or a compound file that includes multiple, fixed sequences of data bytes." Dec. 10-11 (citing Ex. 1001, 5:44-54). The focus of the dispute between EMC and PersonalWeb is not whether a single binary in Woodhill constitutes a single, fixed sequence of data bytes, or whether multiple binary objects in Woodhill constitute multiple, fixed sequence of data bytes. Instead, the focus of the dispute between EMC and PersonalWeb is whether Woodhill's binary object constitutes a "named data item," as required by our claim construction of the claim term "data file."

We are not persuaded by PersonalWeb's argument that Woodhill's binary object is not a claimed "data file" because the binary object is not a "named data item." Woodhill explicitly discloses "data files comprised of *one or more* binary objects." Ex. 1005, 2:3 (emphasis added). Independent claim 1 of Woodhill further recites a "means for dividing each data file into *one or more binary* objects of a predetermined size." Ex. 1005, 21:64-65 (emphasis added). Woodhill also discloses that, if a data file is less than one megabyte, then a single binary object represents a data file. *See, e.g.*, Ex. 1003, 4:23-26; Ex. 1078 ¶ 5. Finally, as we discussed previously, Woodhill discloses that Binary Object Identifier 74 uniquely identifies the binary object associated therewith. Ex. 1005, 4:45-47, 8:33-65. Therefore, consistent with our claim construction of the claim term "data file," a single binary object in Woodhill constitutes a claimed "data file" because it constitutes a data item that is identified uniquely by its corresponding Binary Object Identifier 74.

For instance, in the scenario where a single binary object in Woodhill constitutes a claimed "data file," Dr. Clark confirms that that such a binary object would be named by its corresponding Binary Object Identifier 74, in addition to its File Name 40 as illustrated in Figure 3 of Woodhill. Ex. 1078 ¶ 7. We credit Dr. Clark's testimony because it is consistent with Woodhill's disclosure that a data file may include one binary object that is identified uniquely by its corresponding Binary Object Identifier 74 (Ex. 1005, 2:3, 4:45-47, 8:33-65), as well as Woodhill's disclosure regarding File Name 40 (Ex. 1005, 3:61). Based on the above-identified disclosures in Woodhill, as well as Dr. Clark's supporting testimony, we agree with EMC that a single binary object in Woodhill constitutes the claimed "data file."

We are not persuaded by PersonalWeb's argument that, according to the specification of the '280 patent, the claimed "a hash of the contents of the data file" requires that the hash must be of *all the data* in the data file. Ex. 1001, 43:63 (emphasis added). PersonalWeb's argument is not commensurate in scope with independent claim 36. Independent claim 36 simply recites "the request including a hash of the contents of the data file." It does not indicate explicitly whether the claimed "hash" must be of all the data in the data file. To support its construction that the claimed "hash of the contents of the data file" requires that the hash must be of all the data in the data file, PersonalWeb directs us to the disclosure in the specification of the '280 patent regarding substantially unique identifiers, otherwise referred to as True Names, for a data item. *See, e.g.*, Ex. 1001, 1:14-16, 3:29-31, 33:1-7. However, we note that independent claim 36 does not recite a "substantially unique identifier," or a "True Name," and, therefore, it would be improper for us to read the requirements of these terms from the specification into independent claim 36. *See In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993).

As we discussed previously, during Woodhill's self-auditing procedure, Distributed Storage Manager program 24 uses Binary Object Identifier 74 to identify and request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25. *See* Ex. 1005, 18:16-19. Given that Woodhill's Binary Object Identifier 74 includes Binary Object Hash field 70, such a request necessarily encompasses a hash of contents of the binary object, itself. Ex. 1005, 7:64-8:32. Based on those cited disclosures, we agree with EMC that Woodhill's self-auditing procedure, which includes using Binary Object

Hash field 70 in Binary Object Identifier 74 to identify and request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25, describes "[a] request including a hash of the contents of the data file," as recited in independent claim 36.

Nonetheless, even if we were to assume that the claimed "a hash of the contents of the data file" requires that the hash must be of all the data in the data file, Woodhill still discloses a scenario that would satisfy such a requirement. In light of our analysis above, there is sufficient evidence to support a finding that a single binary object in Woodhill constitutes a claimed "data file." *See* Ex. 1005, 2:3, 4:23-26, 21:64-65. In that scenario, Woodhill would hash all the data in the data file simply by processing one binary object.

c. A single binary object in Woodhill constitutes the claimed "data file"

In its Patent Owner Response, PersonalWeb contends that Woodhill fails to disclose a named "data file" that consists of only one binary object. PO Resp. 8. PersonalWeb argues that, even if a data file in Woodhill includes only one binary object, this does not mean necessarily that the binary object makes up the entire data file. *Id*. (citing Ex. 2013 ¶ 107). PersonalWeb argues that data in one of Woodhill's data files may very well include both metadata and the binary object. *Id*. (citing Ex. 1005, 4:18-19). Therefore, PersonalWeb asserts that, in the scenario where a data file in Woodhill includes both metadata and the binary object, the hash of that binary object does not include necessarily a "hash of the contents of the data file," as required by independent claim 36. *Id*. at 8-9.

In Reply, EMC contends that Woodhill discloses that a named "data file" may consist of only one binary object. Reply 12 (citing Ex. 1005, 1:66-2:3, 3:54-61, 21:64-65). In that scenario, EMC argues that Woodhill's Binary Object Identifier 74 necessarily would include a hash of the contents of the binary object or data file associated therewith. *Id*. (citing Ex. 1005, 8:58-60). According to EMC, Dr. Clark confirms that Woodhill discloses files having a single binary object where the hash of the binary object is a hash of the entire contents of the file. *Id*. (citing Ex. 1078 ¶¶ 4-6).

We are not persuaded by PersonalWeb's argument that Woodhill fails to disclose a named "data file" that consists of only one binary object. As we explained previously, Woodhill discloses at least one scenario where a data file consists of a single binary object. Woodhill explicitly discloses "data files comprised of *one or more* binary objects." Ex. 1005, 2:3 (emphasis added). Independent claim 1 of Woodhill further recites a "means for dividing each data file into *one or more binary* objects of a predetermined size." Ex. 1005, 21:64-65 (emphasis added). Woodhill also discloses that, if a data file is less than one megabyte, then a single binary object represents a data file. *See, e.g.,* Ex. 1003, 4:23-26; Ex. 1078 ¶ 5. Based on these cited disclosures in Woodhill, we agree with EMC that a single binary object in Woodhill constitutes a claimed "data file."

We also are not persuaded by PersonalWeb's argument that, in the scenario where a data file in Woodhill includes both metadata and a single binary object, the hash of that binary object does not include necessarily a "hash of the contents of the data file," as required by independent claim 36. PersonalWeb's argument is predicated on the notion that the claimed "a hash of the contents of the data file" requires that the hash must be of all the data in the data file. However, as we explained previously, the claimed "a hash of the contents of the data file" does not require that the hash must be of all the data in the data file. Moreover, PersonalWeb's proposed scenario where a data file in Woodhill includes both metadata and the binary object is merely an example. The relevant disclosure is Woodhill states: "*For example*, a file *may* contain its normal data and *may also* contain extended attribute data." Ex. 1005, 4:18-19 (emphasis added). The use of permissive terms, such as "for example" and "may," clearly indicates that a data file in Woodhill is not required to include both metadata and a single binary object.

Nonetheless, even if we were to assume that the claimed "a hash of the contents of the data file" requires that the hash must be of all the data in the data file, Woodhill still discloses a scenario that would satisfy such a requirement. In light of our analysis above, there is sufficient evidence to support a finding that a single binary object in Woodhill, which does not include additional metadata, constitutes a claimed "data file." Ex. 1005, 2:3, 4:23-26, 21:64-65. In that scenario, Woodhill would hash all the data in the data file simply by processing one binary object.

d. Woodhill does not provide a lexicographic definition for a data file that indicates it includes at least two data streams or binary objects

In its Patent Owner Response, PersonalWeb contends that Woodhill explicitly defines a data file that is subject to both the backup procedure and the self-auditing procedure as requiring at least two data streams or binary objects. PO Resp. 9 (citing Ex. 1005, 4:15-23; Ex. 2013 ¶ 108.) PersonalWeb asserts that the Board agreed in the Decision to Institute that Woodhill defines a data file in this manner. *Id.* (citing Dec. 14). Using this explicit or special definition of a data file in Woodhill, PersonalWeb argues that Woodhill fails to disclose applying a hash to a combination of at least two binary objects of a file. *Id*. at 10. PersonalWeb then asserts that Woodhill fails to disclose the claimed "hash of the contents of a data file" because each file backed up in Woodhill has at least two binary objects, and Woodhill fails to disclose applying a hash to a combination of binary objects. *Id*.

In its Reply, EMC contends that PersonalWeb's argument that Woodhill's data file must have two data streams or binary objects is misplaced. Reply 11. In particular, EMC argues that Woodhill does not provide an explicit or special definition for a data file that requires it to have at least two data streams or binary objects. *Id*. EMC also argues that there is no basis for PersonalWeb's argument that the Board agreed in the Decision to Institute that Woodhill defines a data file in that manner. *Id*. (citing Dec. 14.) Instead, EMC argues that the Board, in the Decision to Institute, simply reiterated the actual text of Woodhill and never suggested that it qualified as an explicit or special definition for a data file. *Id*.

We are not persuaded by PersonalWeb's argument that Woodhill provides an explicit or special definition for a data file that requires it to have at least two data streams or binary objects. The relevant portion of Woodhill's disclosure is reproduced below.

> The Distributed Storage Manager program 24 views a file as collection of data streams. A data stream is defined as a distinct collection of data within the file that may be changed independently from other distinct collections of data with the file. . . . The Distributed Storage Manager program 24 further divides each data stream into one or more binary objects.

Ex. 1005, 4:13-23.

Although Woodhill discloses that Distributed Storage Manager program 24 views a file as a collection of data streams, it does not set forth a definition for a data file with reasonable clarity, deliberateness, and precision. *See In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). In other words, Woodhill does not define explicitly a data file as requiring at least two data streams or binary objects. To the contrary, Woodhill discloses on at least two occasions that a data file may consist of only one binary object. *See, e.g.*, Ex. 1005, 2:3, 21:64-65.

We also disagree with PersonalWeb's assertion that we explicitly defined a data file in the Decision on Institute as requiring at least two data streams or binary objects. When providing a general summary of Woodhill's disclosure, we simply reiterated the relevant disclosure in Woodhill reproduced above. Dec. 14 (citing Ex. 1005, 13-30). We did not state, nor did we suggest, that Woodhill provides an explicit or special definition for a data file. PersonalWeb's allegation to the contrary is a mischaracterization of our Decision to Institute.

In summary, we maintain that Woodhill's self-auditing procedure, which includes using Binary Object Hash field 70 in Binary Object Identifier 74 to identify and request a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25, describes "[a] request including a hash of the contents of the data file," as recited in independent claim 36. For the foregoing reasons, we conclude that EMC has demonstrated by a preponderance of the evidence that independent claim 36 is anticipated by Woodhill.

4. Independent Claim 38

PersonalWeb relies upon essentially the same arguments presented against independent claim 36 to rebut the explanations provided by EMC as to how Woodhill describes independent claim 38. PO Resp. 11. For the same reasons discussed above with respect to independent claim 36, PersonalWeb's arguments are not persuasive. Therefore, we conclude that EMC has demonstrated by a preponderance of the evidence that independent claim 38 is anticipated by Woodhill.

D. Obviousness over Woodhill—Independent Claims 36 and 38

EMC contends that independent claims 36 and 38 are unpatentable under § 103(a) over Woodhill. Pet. 47-48. In support of that alleged ground of unpatentability, EMC provides explanations as to how Woodhill teaches or suggests each claim limitation. *Id*. (citing Ex. 1032). EMC also submits the declaration of Dr. Clark (Ex. 1009 ¶¶ 28-29) to support its positions. Upon reviewing EMC's Petition and supporting evidence, as well as PersonalWeb's Patent Owner Response and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that independent claims 36 and 38 are obvious over Woodhill.

We begin our analysis with the principles of law that generally apply to a ground of unpatentability based on obviousness, and then we turn to the arguments presented by both EMC and PersonalWeb that are directed to whether Woodhill, as a whole, would have taught or suggested "storing copies of the data file on a set of servers in the network distinct from the first server," as recited in independent claims 36 and 38, to one with ordinary skill in the art.

1. Principles of Law

A patent claim is unpatentable under § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc*., 550 U.S. 398, 406 (2007).
The question of obviousness is resolved on the basis of underlying factual determinations, which include the following: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). We analyze the ground of unpatentability based on obviousness over Woodhill with the above-identified principles in mind.

2. PersonalWeb's Contentions

a. There are no deficiencies in Woodhill to cure

At the outset, PersonalWeb contends that EMC's contentions regarding obviousness do not cure the deficiencies in Woodhill that are discussed above with respect to independent claims 36 and 38. PO Resp. 11. As we explained in our discussion of the ground of unpatentability based on anticipation by Woodhill, there are no such deficiencies in Woodhill to cure.

b. Woodhill, as a whole, would have taught or suggested the claimed "storing copies of the data file on a set of servers in the network distinct from the first server" to one with ordinary skill in the art

Independent claims 36 and 38 both recite, in relevant part, "storing copies of the data file on a set of servers in the network distinct from the first server." Ex. 1001, 43:60-61, 44:7-9.

In its Petition, EMC contends that, to the extent PersonalWeb asserts that Woodhill does not disclose the claimed "storing the data file is [sic] on a first server and storing copies of the data file on a set of servers [in the network] distinct from the first server," a person of ordinary skill in the art would have found it obvious to modify Woodhill to satisfy that claim limitation. Pet. 47 (citing Ex. 1009 ¶¶ 28, 29). EMC argues that distributing files in a network that includes many servers was old and well known in the art. *Id*. EMC further argues that it would have been obvious to one with ordinary skill in the art to add a remote backup file server or servers to Woodhill's system for additional data security, e.g., in the event that remote backup file server 12 is destroyed concurrently with local computers 20 on which a binary object is backed up. *Id*. (citing Ex. 1009 ¶ 29). According to EMC, Dr. Clark confirms that adding a remote backup file server or servers to Woodhill's system would constitute applying a known technique, such as adding extra redundancy, to a known device ready for improvement to yield predictable results. *Id*.

In its Patent Owner Response, PersonalWeb contends that, even if Woodhill was modified to add a remote backup file server or servers, as asserted by EMC, there would have been no logical reason to make such a modification. PO Resp. 11-12. In its Reply, EMC reiterates that Dr. Clark confirms that it would have been well within the routine creativity of one with ordinary skill in the art to add a remote backup file server or servers to the system disclosed in Woodhill. Reply 12 (citing Ex. 1009 ¶ 29).

As illustrated in Figure 1 of Woodhill, which was reproduced previously, network computer system 10 includes remote backup file server 12 and local computers 20 connected via multiple local area networks 16. Ex. 1005, 3:6-31. Woodhill further discloses storing a copy of each binary object in the following three locations: (1) on local computer 20; (2) on another local computer 20 other than local computer 20 where the binary object originally resided; and (3) on remote backup file server 12. Ex. 1005, 9:30-38. Based on these cited disclosures, Woodhill describes "storing the data file is [sic] on a first server and storing copies of the data file on a set of servers in the network distinct from the first server," as recited in independent claims 36 and 38.

Nonetheless, even if we assume that Woodhill does not disclose the claimed "storing copies of the data file on a set of servers in the network distinct from the first server," we agree with EMC that the distribution of binary objects or files in a network computer system containing multiple servers is both old and well known in the art. According to Dr. Clark, a person of ordinary skill in the art would have found it obvious to add a remote backup file server or servers to Woodhill's system for additional data security, such that if Woodhill's remote backup file server 12 is destroyed along with local computers 20, copies of each binary object or file may be preserved on the newly added remote backup file server or servers. *See* Ex. 1009 ¶ 29 (citing Ex. 1005, 9:40-45). In our view, such a modification to Woodhill amounts to nothing more than the combination of familiar elements according to a known method that predictably would result in ensuring that at least one copy of each binary object or file is preserved and not destroyed. *See KSR*, 550 U.S. at 416.

In summary, EMC has presented sufficient evidence that Woodhill, as a whole, would have taught or suggested "storing copies of the data file on a set of servers in the network distinct from the first server," as recited in independent claims 36 and 38, to one with ordinary skill in the art.

c. Secondary Considerations of Non-Obviousness—Licenses

In its Patent Owner Response, PersonalWeb contends that third parties have licensed the '791 patent and any continuations thereof, which includes the '280 patent at issue in this proceeding. PO Resp. 12. PersonalWeb argues that, because third parties have licensed these patents, evidence of non-obviousness exists that outweighs the evidence of obviousness based on Woodhill presented by EMC this proceeding. *Id*. In support of its argument, PersonalWeb directs us to three licensing agreements (Exs. 2010-12), as well as the declaration of Kevin Bermeister (Ex. 2009 ¶¶ 3-9), and then argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between independent claims 36 and 38 and the above-identified licensing agreements. Reply 12. EMC argues that PersonalWeb does not provide any evidence that independent claims 36 and 38 motivated the decision to grant these licensing agreements, and each of the three licenses involved related parties with interlocking ownership and business interests. *Id*. at 12-13. We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and the claimed subject matter recited in independent claims 36 and 38.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *GPAC*, 57 F.3d at 1580. Further, without a showing of nexus, "the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness" when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000); *see Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus in two respects. First, neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010-12) are directed to the claimed subject matter recited in independent claims 36 and 38. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in independent claims 36 and 38. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Second, even if we assume that the above-identified licenses establish some degree of industry respect for the claimed subject matter recited in independent claims 36 and 38, that success is outweighed by evidence of obviousness over Woodhill discussed above.

Based on the record before us, including the evidence of obviousness based on Woodhill and the evidence of secondary considerations regarding licensing activities, we conclude that EMC has demonstrated by a preponderance of the evidence that independent claims 36 and 38 are obvious over Woodhill.

E. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following evidence: (1) paragraph 10 of the rebuttal declaration of Dr. Clark that relies on, and cites to, Langer because it is irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding; (2) Langer, because it is not authenticated properly under Federal Rule of Evidence ("FRE") 901; (3) Langer, because it includes impermissible hearsay, in violation of FRE 802; and (4) paragraphs 7 and 13 of the rebuttal declaration of Dr. Clark because he relies upon subject matter in Woodhill that does not qualify as prior art to the '280 patent. Paper 60 ("PO Mot."). EMC opposes PersonalWeb's motion to exclude. Paper 69 ("Pet. Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 73 ("PO Reply"). For the reasons discussed below, PersonalWeb's motion to exclude is denied.

1. The statements in Dr. Clark's rebuttal declaration regarding Langer are admissible evidence

PersonalWeb contends that paragraph 10 of the rebuttal declaration of Dr. Clark (Ex. 1078) should be excluded because this paragraph relies upon, and cites to, Langer (Ex. 1003). PO Mot. 1. PersonalWeb argues that this proceeding was instituted based only on Woodhill—not on Langer—and, therefore, EMC's reliance on Langer is outside the scope of this proceeding and impermissible. *Id*. In response, EMC contends that Dr. Clark's testimony regarding Langer was offered in response to PersonalWeb's argument that Woodhill does not disclose a "client request" including a hash of the contents of a data file. Pet. Opp. 1-2. EMC also argues that Dr. Clark's testimony serves to corroborate the state of the art at the time of the '280 patent, as well as the requisite detail needed for such a basic computer operation. In reply, PersonalWeb contends that EMC improperly relies on Langer as alleged prior art and attempts to shoehorn into the record additional teachings not disclosed or suggested in Woodhill. PO Reply 1.

We agree with EMC that it may rely upon Langer to corroborate the state of the art at the time of the '280 patent, as well as the requisite detail needed for a basic computer operation. The '280 patent has an effective filing date of April 11, 1995. Ex. 1001 at [62]. Langer has a publication date of August, 7, 1991. Ex. 1003, 1. Therefore, Langer has a publication date prior to April 11, 1995. We recognize that Langer was relied on by EMC's rebuttal declarant, Dr. Clark, to indicate that it was old and well known to request binary objects using their identifiers (Ex. 1078 ¶ 10), and it is the type of document that experts in the pertinent field reasonably would rely on to formulate their opinions. In other words, EMC may rely on Langer to demonstrate what one with ordinary skill in the pertinent art would have known about basic computer operations at the time of the '280 patent, such as a client request that includes an identifier.

For the foregoing reasons, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude paragraph 10 of the rebuttal declaration of Dr. Clark.

2. EMC provides sufficient evidence to support a finding that Langer has been authenticated properly

PersonalWeb contends that EMC fails to provide evidence indicating that Langer existed prior to the effective filing date of the ’280 patent—April 11, 1995—and, therefore, should be excluded under FRE 901. PO Mot. 2. In particular, PersonalWeb argues that Langer allegedly was downloaded from the Internet in 2003 based on the “7/29/2003” date in the lower, right-hand corner. *Id*. PersonalWeb also argues that authentication of Langer requires personal knowledge of its existence prior to April 11, 1995. *Id*. at 3. In response, EMC contends that it submitted sworn testimony from Mr. Keith Moore that properly authenticates Langer under FREs 901(b)(1) and (4), 901(b)(3), 901(b)(8), and 901(b)(6) and (7). Pet. Opp. 2-4 (citing Ex. 1048 ¶¶ 5-11). In reply, PersonalWeb contends that Langer is not authenticated properly under the FREs identified by EMC. PO Reply. 1-5.

We agree with EMC that Langer has been authenticated properly under FRE 901(b)(1) and (4) because Mr. Moore testified that Langer is an article posted on Usenet newsgroups on August 7, 1991 (Ex. 1048 ¶¶ 11-15), and it includes distinct header fields unique to Usenet formatting and content (*id*. at ¶¶ 16,17). Although PersonalWeb presents several theories that attack the authenticity of Langer, PersonalWeb fails to explain adequately why the testimony offered by Mr. Moore does not authenticate Langer. PersonalWeb simply presents attorney arguments and does not offer testimony from its own expert that is contrary to the testimony offered by Mr. Moore. Therefore, based on the record before us, EMC has presented sufficient evidence to support a finding that Langer has been authenticated properly under FRE 901(b)(1) and (4).

We also are not persuaded by PersonalWeb's argument that the download date of "7/29/2003" in the lower, right-hand corner calls into question whether Langer existed prior to April 11, 1995. The mere fact that a "downloaded" copy of Langer has a date subsequent to the earliest effective filing date is not sufficient to rebut EMC's supporting evidence that Langer is what it claims to be—namely an article posted on Usenet newsgroups on August 7, 1991. *See, e.g.*, Ex. 1048 ¶¶ 11-17.

To the extent PersonalWeb argues that Mr. Moore cannot authenticate Langer because he does not have personal knowledge of its existence prior to April 11, 1995, or that Mr. Albert Langer is the only person that can authenticate Langer properly, we disagree. Neither a declaration from Mr. Langer, nor evidence of someone actually viewing Langer prior to April 11, 1995, is required to support a finding that Langer is what it claims to be. *See In re Wyer*, 655 F.2d 221, 226 (CCPA 1981) (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'"); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would have known of the invention.).

For the foregoing reasons, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Langer as unauthenticated evidence.

3. Langer is not inadmissible hearsay

PersonalWeb contends that the dates in Langer, or any other information that purports to establish a publication date for Langer, are inadmissible hearsay under FRE 802 and not subject to any hearsay exception. PO Mot. 3-4. PersonalWeb also argues that, to the extent that EMC contends that any statements in Langer were made prior to the critical date of the '280 patent, the entirety of Langer is inadmissible hearsay. *Id*. at 4. In response, EMC contends that Langer is not hearsay because it is being offered for what it describes—not for the truth of its disclosure. Pet. Opp. 4. EMC also argues that the August 7, 1991, posting date on Langer's header and its uniform resource locator ("URL") both were generated automatically by the hosting computer and, therefore, are admissible as non-hearsay to prove Langer's August 1991 publication date. *Id*. at 4-5 (citing Ex. 1048 ¶ 7). In reply, PersonalWeb maintains that the dates and other information in Langer used to establish its availability as of August 1991 amount to inadmissible hearsay. PO Reply 5.

We recognize that EMC's rebuttal declarant, Mr. Moore, reasonably would rely on the date of August 7, 1991, that appears in both Langer's header and URL to formulate his opinion on whether Langer was available publicly as of that date. Accordingly, the date of August 7, 1991, posted in Langer need not be admissible for the testimony of Mr. Moore to be admissible. Nonetheless, we agree with EMC that the date of August 7, 1991, posted on Langer's header and URL serve a non-hearsay purpose for which it can be admitted—namely to prove that the document was available publicly as of that date.

Moreover, we are not persuaded by PersonalWeb's arguments that Langer, in its entirety, constitutes hearsay. With the exception of the dates in Langer, PersonalWeb does not identify specifically the textual portions of Langer that allegedly are being offered for the truth of the matter asserted, yet does seek to exclude Langer in its entirety. We will not go through the entirety of Langer and determine which portions PersonalWeb believes to be hearsay—this is something that PersonalWeb should have done in its motion to exclude.

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude the dates posted in Langer, or any statements made therein, as impermissible hearsay.

4. The statements in Dr. Clark's rebuttal declaration that rely on the claim language of Woodhill are admissible

PersonalWeb contends that paragraphs 7 and 13 of Dr. Clark's rebuttal declaration (Ex. 1078) that rely upon, and cite to, the claims of Woodhill should be excluded as irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding. PO Mot. 4. In particular, PersonalWeb argues that the "name" of a particular binary object identifier, as recited in the claims of Woodhill, is not prior art to the '280 patent because there is insufficient written description support in Woodhill's original disclosure for that claimed subject matter. *Id.* at 5. In response, EMC contends that Woodhill's specification provides sufficient written description support for the "name" of a particular binary object identifier, as recited in the claims of Woodhill. Pet. Opp. 6 (Ex. 1005, 7:60-8:65, 18:16-23, fig. 3).

Contrary to PersonalWeb's argument, Woodhill's original disclosure contains sufficient written description support for the "name" of a particular binary object identifier, as recited in the claims of Woodhill. Upon reviewing the description of Binary Object Identification record 58 in Woodhill's original disclosure, the only part of the record that identifies uniquely the binary object associated therewith is Binary Object Identifier 74. Ex. 2007, 26, 33-34.[2] During Woodhill's self-auditing procedure, Distributed Storage Manager program 24 uses Binary Object Identifier 74 to access a randomly selected binary object by retrieving its corresponding Binary Object Identification record 58 in File Database 25. *See* Ex. 2007, 53. Dr. Clark confirms such an operation was routine because it was old and well-known to access records stored in a database using their identifiers. *See* Ex. 1078 ¶¶ 10, 11.

Based on the cited portions in Woodhill's original disclosure, as well as Dr. Clark's corroborating testimony, we are persuaded that Woodhill's original disclosure conveys with reasonable clarity to one with ordinary skill in the art that Binary Object Identifier 74 may be considered a "name" for a binary object associated therewith because it uniquely identifies that binary object. *See Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc) (The written description test is whether the original disclosure of the application relied upon reasonably conveys to a

[2] Exhibit 2007 includes excerpts from the file history of Woodhill. PersonalWeb did not provide any page numbers for this Exhibit. For purposes of this decision, page 1 is the page that includes "Exhibit PersonalWeb 2007" in the lower, right-hand corner. The remaining pages are numbered consecutively therefrom.

person of ordinary skill in the art that the inventor had possession of the claimed subject matter as of the filing date.)

Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude paragraphs 7 and 13 of Dr. Clark's rebuttal declaration that rely upon, and cite to, the "name" of a particular binary object identifier, as recited in the claims of Woodhill.

F. EMC's Motion to Exclude

EMC seeks to exclude three license agreements (Exs. 2010-12), as well as the two declarations offered by Mr. Kevin Bermeister relating to those license agreements (Exs. 2009, 2014), because they are irrelevant under FRE 401, highly prejudicial, confusing, and misleading under FRE 403. Paper 66. PersonalWeb opposes EMC's motion to exclude. Paper 70. In response, EMC filed a reply to PersonalWeb's opposition to its motion to exclude. Paper 72.

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding the three license agreements (Exs. 2010-12) and the two declarations offered by Mr. Bermeister (Exs. 2009, 2014), we have concluded that EMC has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable. Accordingly, EMC's motion to exclude evidence is dismissed as moot.

III. CONCLUSION

EMC has demonstrated by a preponderance of the evidence that independent claims 36 and 38 of the '280 patent are unpatentable based on the grounds of unpatentability set forth in the table below.

Claims	Basis	Reference
36 and 38	§ 102(e)	Woodhill
36 and 38	§ 103(a)	Woodhill

IV. ORDER

In consideration of the foregoing, it is

ORDERED that, based on a preponderance of the evidence, independent claims 36 and 38 of the ’280 patent are unpatentable;

FURTHER ORDERED that PersonalWeb’s motion to exclude evidence is DENIED;

FURTHER ORDERED that EMC’s motion to exclude evidence is DISMISSED as moot; and

FURTHER ORDERED that, because this is a final written decision, parties to this proceeding seeking judicial review of our decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

For PETITIONERS:

Peter Dichiara
David L. Cavanaugh
WILMER CUTLER PICKERING HALE AND DORR LLP
Peter.Dichiara@wilmerhale.com
David.Cavanaugh@wilmerhale.com

For PATENT OWNERS:

Joseph A. Rhoa
Updeep S. Gill
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

Trials@uspto.gov
571-272-7822

Paper 64
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION,
Petitioner,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00084
Patent 7,945,544 B2

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

CHANG, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

EMC Corporation ("EMC") filed a petition on December 16, 2012, requesting an *inter partes* review of claim 1 of U.S. Patent No. 7,945,544 B2 ("the '544 patent"). Paper 3 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") filed a patent owner preliminary response. Paper 9 ("Prelim. Resp."). Taking into account the patent owner preliminary response, the Board determined that the information presented in the petition demonstrated that there was a reasonable likelihood that EMC would prevail with respect to claim 1. Pursuant to 35 U.S.C. § 314, the Board instituted this trial on May 17, 2013, as to claim 1 of the '544 patent. Paper 14 ("Dec.").

After institution, PersonalWeb filed a patent owner response (Paper 33 ("PO Resp.")), and EMC filed a reply to the patent owner response (Paper 40 ("Reply")). Oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is entered pursuant to 35 U.S.C. § 318(a). We hold that claim 1 of the '544 patent is unpatentable under 35 U.S.C. §§ 102 and 103.

[1] This proceeding, as well as IPR2013-00082, IPR2013-00083, IPR2013-00085, IPR2013-00086, and IPR2013-00087, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 63.

A. Related Proceeding

EMC indicates that the ’544 patent is the subject of litigation titled *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED (E.D. Tex.). Pet. 1.

B. The ’544 patent

The ’544 patent relates to a method for identifying a data item (*e.g.*, a data file or record) in a data processing system, by using an identifier that depends on all of the data in the data item and only on the data in the data item. Ex. 1001, 1:45-49; 3:53-56. Thus, the identity of a data item is said to be independent of its name, origin, location, and address. *Id*. at 3:56-59. According to the ’544 patent, it is desirable to have a mechanism for identifying identical data items to reduce duplicate copies of a data item. *Id*. at 3:37-40. Figure 10(b) of the ’544 patent, reproduced below, is a flow chart for determining an identifier of a simple or compound data item.

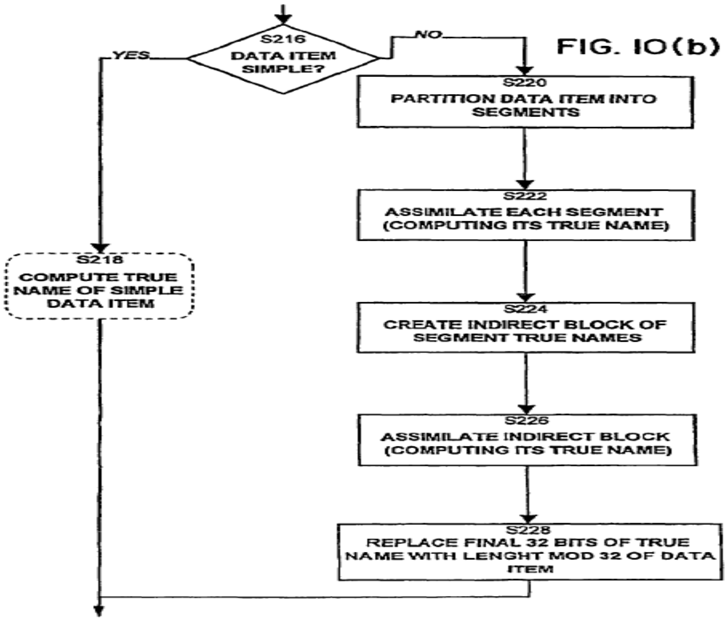


FIG. IO(b)
S216
DATA ITEM SIMPLE?
YES
NO
S218
COMPUTE TRUE NAME OF SIMPLE DATA ITEM
S220
PARTITION DATA ITEM INTO SEGMENTS
S222
ASSIMILATE EACH SEGMENT (COMPUTING ITS TRUE NAME)
S224
CREATE INDIRECT BLOCK OF SEGMENT TRUE NAMES
S226
ASSIMILATE INDIRECT BLOCK (COMPUTING ITS TRUE NAME)
S228
REPLACE FINAL 32 BITS OF TRUE NAME WITH LENGHT MOD 32 OF DATA ITEM

As shown in Figure 10(b) of the '544 patent, for a simple data item (a data item whose size is less than a particular given size) (S216 and S218), a data identifier (True Name) is computed using a function (*e.g.*, a message digest ("MD") function, such as MD4 or MD5, or a secure hash algorithm ("SHA") function). *Id.* at 12:18-49, 13:31-42; figs. 10(a) & 10(b). As a result, a data item that has an arbitrary length is reduced to a relatively small, fixed size identifier (True Name) that represents the data item. *Id.*

If the data item is a compound data item (a data item whose size is greater than the particular given size), the system will partition the data item into segments (S220); assimilate each segment (S222); compute the True Name of the segment; create an indirect block consisting of the computed segment True Names (S224); assimilate the indirect block (S226); and replace the final 32 bits of the resulting True Name by the length modulo 32 of the compound data item (S228). *Id.* at 13:43-61, fig. 10(b). The result is the True Name of the compound data item. *Id.*

Figure 11 of the '544 patent is reproduced below:

FIG. 11

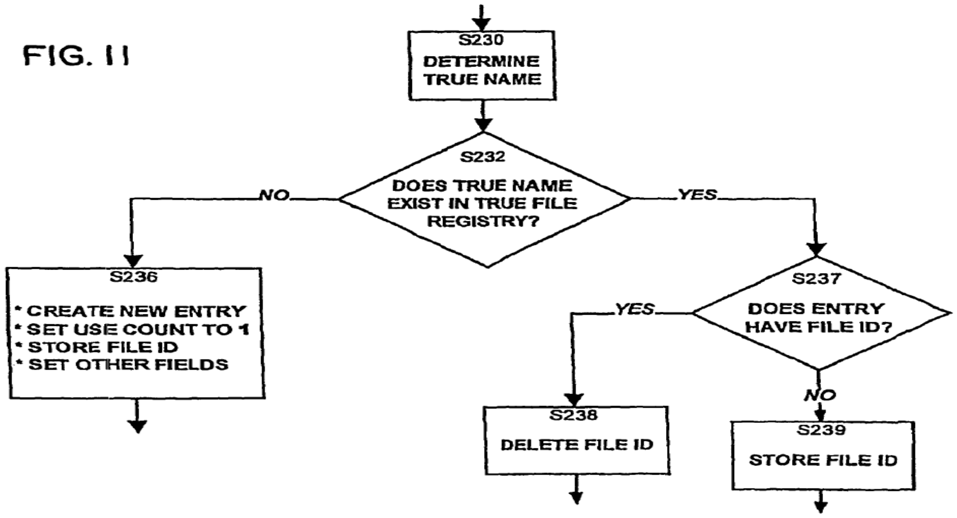


Figure 11 of the ’544 patent depicts a mechanism for assimilating a data item into a file system. The purpose of this mechanism is to add a given data item to the True File registry. *Id*. at 14:4-11. If the data item already exists in the registry, the duplicate will be eliminated. *Id*.

To assimilate a data item, the system will determine the True Name of the data item corresponding to the file (S230); look for an entry for the True Name in the True File Registry (S232); and determine whether a True Name entry exists in the True File Registry (S232). *Id*. at 14:4-27, fig. 11. If the entry record includes a corresponding True File ID (Step S237), the system will delete the file (Step S238). Otherwise the system will store the True File ID in the entry record (S239). *Id*. If there is no entry in the True File Registry for the True Name (S232), the system will create a new entry in the True File Registry for the True Name (S236). *Id*.

C. Challenged Claim

According to EMC, claim 1 essentially requires obtaining “values” for two data items, and then comparing these values to ascertain whether the two data items correspond to each other (e.g., whether they are the same). Pet. 16. Claim 1 recites the following:

> 1. A computer-implemented method, the method comprising:
>
> (A) for a first data item comprising a first plurality of parts,
>
> (a1) applying a first function to each part of said first plurality of parts to obtain a corresponding part value for each part of said first plurality of parts,

> wherein each part of said first plurality of parts comprises a corresponding sequence of bits, and
>
> wherein the part value for each particular part of said first plurality of parts is based, at least in part, on the corresponding bits in the particular part, and
>
> wherein two identical parts will have the same part value as determined using said first function,
>
> wherein said first function comprises a first hash function; and
>
> (a2) *obtaining a first value for the first data item*, said first value obtained by *applying a second function to the part values* of said first plurality of parts of said first data item, said second function comprising a second hash function;
>
> (B) for a second data item comprising a second plurality of parts,
>
> (b1) applying said first function to each part of said second plurality of parts to obtain a corresponding part value for each part of said second plurality of parts,
>
> wherein each part of said second plurality of parts consists of a corresponding sequence of bits, and
>
> wherein the part value for each particular part of said second plurality of parts is based, at least in part, on the corresponding bits in the particular part of the second plurality of parts; and
>
> (b2) *obtaining a second value* for the second data item by applying said second function to the part values of said second plurality of parts of said second data item; and
>
> (C) ascertaining whether or not said *first data item corresponds to said second data item* based, at least in part, on said first value and said second value.

Ex. 1001, 38:34-39:3 (emphases and indentions added).

D. Prior Art Relied Upon

EMC relies upon the following prior art references:

Woodhill	US 5,649,196[2]	July 15, 1997	(Ex. 1005)

Frederick W. Kantor, "*FWKCS (TM) Contents-Signature System Version 1.22*," FWKCS122.REF (Aug. 10, 1993) ("Kantor," Ex. 1004)

E. Grounds of Unpatentability

The Board instituted the instant trial based on the following grounds of unpatentability:

Claim	Basis	References
1	§ 102(e)	Woodhill
1	§ 102(b)	Kantor
1	§ 103(a)	Kantor and Woodhill

II. ANALYSIS

A. Claim Construction

We begin our analysis by determining the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary

[2] Woodhill claims the benefit of U.S. Patent Application No. 08/085,596, which was filed on July 1, 1993.

meaning as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). An inventor may rebut that presumption by providing a definition of the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). In the absence of such a definition, limitations are not to be read from the specification into the claims. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993).

In the Decision on Institution, we construed the claim term "data item" to mean "sequence of bits," and observed that in the context of the specification, the meaning also includes one of the following: (1) the contents of a file; (2) a portion of a file; (3) a page in memory; (4) an object in an object-oriented program; (5) a digital message; (6) a digital scanned image; (7) a part of a video or audio signal; (8) a directory; (9) a record in a database; (10) a location in memory or on a physical device or the like; and (11) any other entity which can be represented by a sequence of bits. Dec. 9. The parties agree with that claim construction. Pet. 6; PO Resp. 1. As noted in the Decision on Institution, that claim construction is consistent with the specification. Dec. 8-9 (citing Ex. 1001, 2:17-18 ("the terms 'data' and 'data item' as used herein refer to sequences of bits."); *id*. at 2:18-22, 27-32). We discern no reason to deviate from that claim construction for the purposes of this decision.

B. Principles of Law

To establish anticipation, each and every element in a claim, arranged as recited in the claim, must be found in a single prior art reference. *Net MoneyIN, Inc. v. VeriSign, Inc*., 545 F.3d 1359, 1369 (Fed. Cir. 2008); *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001). We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *Paulsen*, 30 F.3d at 1480. Moreover, "it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." *In re Preda*, 401 F.2d 825, 826 (CCPA 1968).

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc*., 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 (1966). The level of ordinary skill in the art is reflected by the prior art of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001);

In re GPAC Inc., 57 F.3d 1573, 1579 (Fed. Cir. 1995); *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978).

We analyze the instituted grounds of unpatentability in accordance with the above-stated principles.

C. Claim 1 – Anticipated by Woodhill

EMC asserts that claim 1 is unpatentable under 35 U.S.C. § 102(e) as anticipated by Woodhill. Pet. 50-57. As support, EMC provides detailed explanations as to how each claim element, arranged as recited in the claim, is disclosed by Woodhill. *Id*. EMC also relies on the declaration of Dr. Douglas W. Clark. Ex. 1009 ¶¶ 43-49.

PersonalWeb counters that Woodhill does not describe all of the limitations of claim 1. PO Resp. 3-15. Specifically, PersonalWeb contends that: (1) Woodhill fails to describe applying a second hash function to shadow files (*id*. at 5-11 (citing Ex. 2016 ¶¶ 25-35)); and (2) Woodhill does not describe binary object identifiers for the first data item and the second data item (*id*. at 11-15 (citing Ex. 2016 ¶¶ 36-40)). PersonalWeb also proffers a declaration of Dr. Robert B. K. Dewar. Ex. 2016 ¶¶ 20-41.

Upon review of the parties' arguments and evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claim 1 is unpatentable under 35 U.S.C. § 102(e) as being anticipated by Woodhill.

<u>Woodhill</u>

Woodhill discloses a system for distributed storage management on a computer network system using binary object identifiers. Ex. 1005, 1:11-17.

The system includes a remote backup file server and a plurality of local area networks in communication with the remote backup file server. *Id*.

Figure 1 of Woodhill, reproduced below, depicts a computer network system that includes a distributed storage management system:

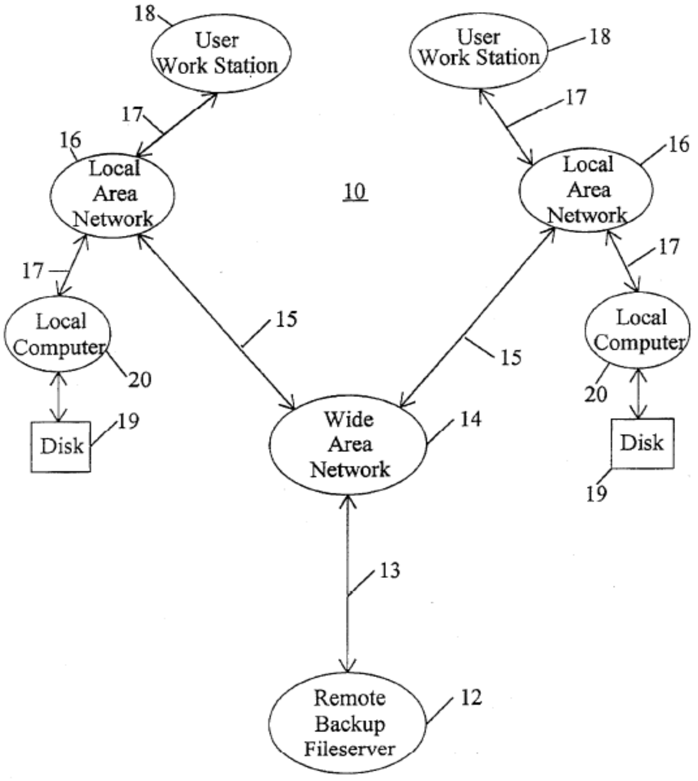


FIG. 1

As illustrated in Figure 1 of Woodhill, remote backup file server 12 communicates with wide area network 14, which communicates with a plurality of local area networks 16. *Id*. at 3:12-30. Each local area network 16 includes multiple user workstations 18 and local computers 20. *Id*. at 3:24-44. The storage space on each disk drive 19 on each local computer 20 is allocated according to the hierarchy illustrated in Figure 2. *Id*. at 3:31-44.

Woodhill's system includes a Distributed Storage Manager (DSM) program for building and maintaining the file database. *Id*. at 3:44-49. The DSM program views a file as a collection of data streams, and divides each data stream into one or more binary objects. *Id*. at 4:13-23; 7:40-43; fig. 5A, item 132. Specifically, data streams represent regular data, extended attribute data, access control list data, etc. *Id*. at 7:44-47. If the size of the data stream is larger than the maximum binary object size, then the DSM program divides the data stream into multiple binary objects; otherwise, a single binary object represents the data stream. *Id*. at 4:23-30; 7:47-59; fig. 5A, items 134 and 136. For each binary object being backed up, a binary object identification record is created in a file database and includes a Binary Object Identifier to identify a particular binary object uniquely. *Id*. at 7:60-8:1; 8:33-34.

Binary object identifiers are calculated based on the contents of the data instead of from an external and arbitrary source so that the binary object identifier changes when the contents of the binary object changes. *Id*. at 8:57-62; 8:40-42. Notably, the binary object identifier includes a binary object hash field that is calculated against the contents of the binary object taken one word (16 bits) at a time using a hash algorithm. *Id*. at 8:22-32. According to Woodhill, duplicate binary objects can be recognized from their identical binary object identifiers, even if the objects reside on different types of computers in a heterogeneous network. *Id*. at 8:62-65.

For large database files on the network computer system, the DSM program utilizes a technique of subdividing the large database files into granules, and then tracks changes from the previous backup copy of the "granule" level. *Id.* at 14:53-65. This technique is used to reduce the amount of data that must be transmitted to the remote backup file server. *Id.* at 15:4-8. Figure 5G of Woodhill illustrates the "granularization" procedure and is reproduced below:

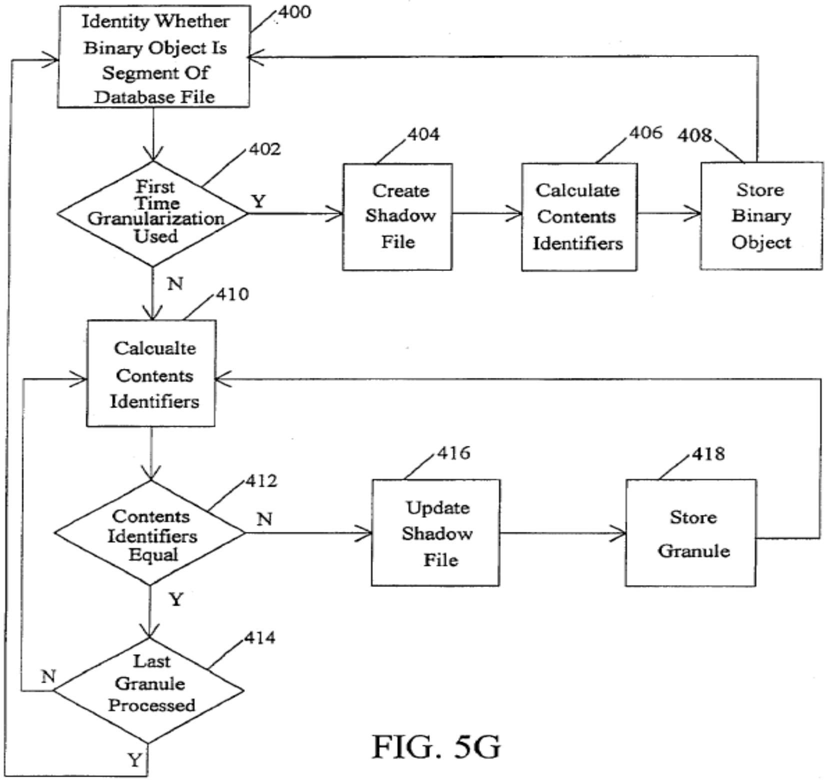

Identity Whether Binary Object Is Segment Of Database File
400
First Time Granularization Used
402
Y
N
Create Shadow File
404
Calculate Contents Identifiers
406
Store Binary Object
408
Calcualte Contents Identifiers
410
Contents Identifiers Equal
412
N
Y
Update Shadow File
416
Store Granule
418
Last Granule Processed
414
N
Y

FIG. 5G

As depicted in Figure 5G, if this is the first time that the binary object is being backed up using the "granularization" technique (step 402), the DSM program creates a shadow file, which contains a contents identifier for each granule in the binary object (step 404). *Id.* at 15:9-24. Each contents identifier includes a 32-bit hash number which is calculated against the contents of the granule. *Id.* at 15:24-30; Fig. 5A, step 138.

Each time that the binary object is backed up, the DSM program calculates the contents identifier for each granule in the binary object, and then compares it to the contents identifier of the granule from the last time the binary object was backed up to determine if the granule has changed. *Id*. at 15:32-38. At step 406, the DSM program calculates a change identifier for each granule of the binary object and stores it in the shadow file for that binary object. *Id*. at 15:40-45.

<u>Applying a second hash function to shadow files</u>

Claim 1 requires "obtaining a first value for the first data item, said first value obtained by applying a second [hash] function to the part values of said first plurality of parts of said first data item" (i.e., "a hash of hashes"). In its petition, EMC asserts that Woodhill's binary object identifiers for the shadow files meet this limitation. Pet. 53-56 (citing Ex. 1005, 5:62-63, 7:60-8:31; 9:6-28; 15:16-24; Ex. 1009 ¶¶ 43-49).

PersonalWeb, however, argues that Woodhill's granularization process does not disclose applying a second hash function to shadow files. PO Resp. 5, 7 (citing Ex. 2016 ¶¶ 29-35). In particular, PersonalWeb and its expert assert that "Binary Object identifiers 74 are *not* mentioned in connection with Woodhill's 'granularization' procedure, and are not used therein." PO Resp. 8 (citing Ex. 2016 ¶ 31). PersonalWeb also maintains that EMC's reliance on Woodhill's statement that "the default operation is to back up *all files on all disk drives 19 on the local computer 20*" (Ex. 1005, 5:62-63) is incorrect because "Woodhill never describes shadow files as being stored on disk drives 19 of local computers 20." *Id*. at 10 (citing

Ex. 1005, 15:4-9; Ex. 2016 ¶ 34). Additionally, PersonalWeb, citing to its expert testimony, alleges that a binary object identifier is not created for a shadow file, because the granularization process, in which the shadow files are created, is not used for backing up copies of binary objects for storage on local computers. *Id*. at 8-9 (citing Ex. 2016 ¶¶ 31-35). PersonalWeb further contends that a shadow file will not be backed up by the DMS program, as a shadow file does not meet Woodhill's definition of a "file" that requires at least two data streams. *Id*. at 11 (citing Ex. 1005, 4:14-15; Ex. 2016 ¶ 35).

In its reply, EMC responds that Woodhill discloses "the application of a hash to the 'contents identifiers' in a shadow file." Reply 1, n.1. Specifically, EMC alleges that Woodhill discloses calculating a binary object identifier for each shadow file when the DSM program backs up the file. *Id*. at 2. EMC also submits that the shadow file's binary object identifier is *for* the associated underlying file or binary object. *Id*. at 6-7. We agree with EMC.

PersonalWeb and its expert testimony narrowly focus on Woodhill's granularization procedure. Notably, Woodhill specifically states that each of the functions performed by the DSM program operates in cooperation with the other functions to form *a unitary computer program*. Ex. 1005, 4:62-5:2; figs. 5a-5l. The disclosure of Woodhill merely divides the DSM program into several distinct functions for explanation purposes. *Id*.

We agree with EMC that Woodhill's "default operation is to back up all files on all disk drives 19 on the local computer 20" and each *shadow file,* like all files stored on disk drives 19, is divided into one or more binary objects to be backed up. Pet. 53 (citing Ex. 1005, 5:62-63); 55 (citing Ex. 1009 ¶¶ 46-48; Ex. 1005, 4:13-34; 5:61-63). As noted by EMC, in the process of backing up shadow files, Woodhill would obtain a first value by calculating a binary object identifier (i.e., applying a second hash function) for each shadow file binary object (i.e., the part values – the first hash). Pet. 55-56 (citing Ex. 1009 ¶¶ 45-48; Ex. 1005, 7:60-8:31; 15:16-24).

EMC's expert, Dr. Clark, testifies:

> 46. Prior to backing up a binary object using the granularization technique for the first time, the local computer storing the binary object creates a "shadow file" containing the granule contents identifiers for each granule of that binary object. (*Id*. at col. 15, ll. 16-24; Ex 1005.) *Woodhill also discloses claim portions [1c] and [1e]*[3] *through his process of creating shadow files on local computers to store the latest granule contents identifiers for granularized binary objects, and then backup these shadow files*. In particular, a shadow file, including each contents identifier for each granule of a binary object, like any file will be divided into one or more Binary Objects. In some cases, due to the concise nature of a shadow file, a shadow file may be backed up using a single binary object.
>
> 47. As I have illustrated, *each shadow file binary object, like all binary objects, has a corresponding Binary Object Identifier*. Further, each Binary Object Identifier includes a hash of the contents of the Binary Object. Consequently, a Binary Object Identifier for a shadow file binary object satisfies these claim elements because it is *a hash (second function) of the*

[3] "Claim portions [1c] and [1e]" refer to steps (a2) and (b2) of claim 1. Ex. 1009 ¶ 16.

> *contents identifiers, or granule hashes (i.e., "part values" of the plurality of parts [granules]).*

Ex. 1009 ¶¶ 46-47 (emphases added).

Upon reviewing the evidence on record, we credit the testimony of Dr. Clark over that of Dr. Dewar. *See Yorkey v. Diab*, 601 F.3d 1279, 1284 (Fed. Cir. 2010) (holding that Board has discretion to give more weight to one item of evidence over another "unless no reasonable trier of fact could have done so"). We find that Dr. Clark's explanations are consistent with Woodhill. *See, e.g.*, Ex. 1005, 4:13-34; 4:62-5:2; 5:61-63; 7:60-8:31; 15:16-24; figs. 5a-5l. On the other hand, Dr. Dewar's testimony (Ex. 2016 ¶ 34) that shadow files are not stored on the local computers contradicts the disclosure of Woodhill that shadow files are created by the DSM program and stored on the disk drives of the local computers. *See, e.g.*, Ex. 1005, 15:21-24 (The DSM program "creates a 'shadow file' which contains a 'contents identifier' for each 'granule' in the binary object."); 5:6-9 (The DSM program "operates in the same fashion on each local computer 20 on the network computer system 10."); 5:7-9; fig. 2, item 24 (The DSM program resides on each disk drive 19 on each local computer 20.); 3:35-49; fig. 3 (The DSM program builds and maintains file database 25, which includes file identification record 34 and binary object identifier 74, on one of disk drives 19 on each local computer 20.); 14:62-65; 15:4-6 (The DSM utilizes the granularization procedure to subdivide large databases files into granules and then tracks changes from the previous backup copy at the granule level to reduce the amount of data that are being transmitted from the local computer to the remote backup file server.).

To substantiate its position that shadow files are not stored on disk drives 19 on local computers 20, PersonalWeb also relies on Woodhill's statement that the granularization "technique of subdividing files into 'granules' . . . is not utilized in making backup copies of [database file] binary objects for storage on local computers." PO Resp. 10 (citing Ex. 1005, 15:4-9). However, such reliance is misplaced. As EMC notes, reading Woodhill's statement in context, the statement merely confirms that, when backing up large database files using the granularization procedure, the system sends the backup copies of *the database files* to a remote server. Reply 3; *see also* Ex. 1005, 14:59-61 ("As a result, in most cases, the entire 'large' database file would have to be backed up to the remote backup file server 12."). PersonalWeb does not point out where the DSM program would execute the granularization procedure to create the shadow files. Nor does it explain sufficiently why the DSM program would not be executing the granularization procedure *on the local computer*. Given the disclosures of Woodhill noted above, we agree with EMC that the DSM program executes the granularization procedure to create shadow files on disk drive 19 of local computer 20, and not on remote backup file server 12. Reply 3.

We also are not persuaded by PersonalWeb's argument and expert testimony that Woodhill sets forth a definition of the word "file" that requires *at least two data streams*, and that the DMS program would not backup a shadow file to create a binary object identifier, because a shadow file does not meet that alleged definition of the word "file." *See* PO Resp. 11; Ex. 2016 ¶¶ 27, 35. PersonalWeb's argument and expert testimony are not consistent with the explicit disclosure of Woodhill. In particular, they ignore the fact that Woodhill specifically uses the word "file" in the term "shadow *file*." They also do not provide sufficient explanation why a *shadow file* cannot have more than one data stream or more than one binary object. In fact, a shadow file is consistent with Woodhill's description of a file. *See* Ex. 1005, 15:21-24 (the DSM program "creates a 'shadow *file*' which contains a 'contents identifier' for each 'granule' in the binary object."); *id*., 4:18-19 ("[A] file may contain its *normal data* and may also contain extended *attribute data*."); *id*., 2:23-24 ("data files comprised of *one or more* binary objects") (Emphases added.). As EMC notes, the actual text in Woodhill that PersonalWeb relies on is not a definition of the word "file," and does not require a file to have *at least two data streams*. Reply 4 (citing Ex. 1005, 4:14-15). Indeed, Woodhill does not preclude a *file* from having *only one* data stream, or *only one* binary object. Ex. 1005, 2:23-24 ("storing data files comprised of *one* or more binary objects"); 4:21-23 (The DMS program "divides each data stream into *one* or more binary objects.") (Emphasis added.).

For the reasons stated above, EMC has demonstrated by a preponderance of the evidence that Woodhill describes applying a second hash function to shadow files (i.e., "a hash of hashes").

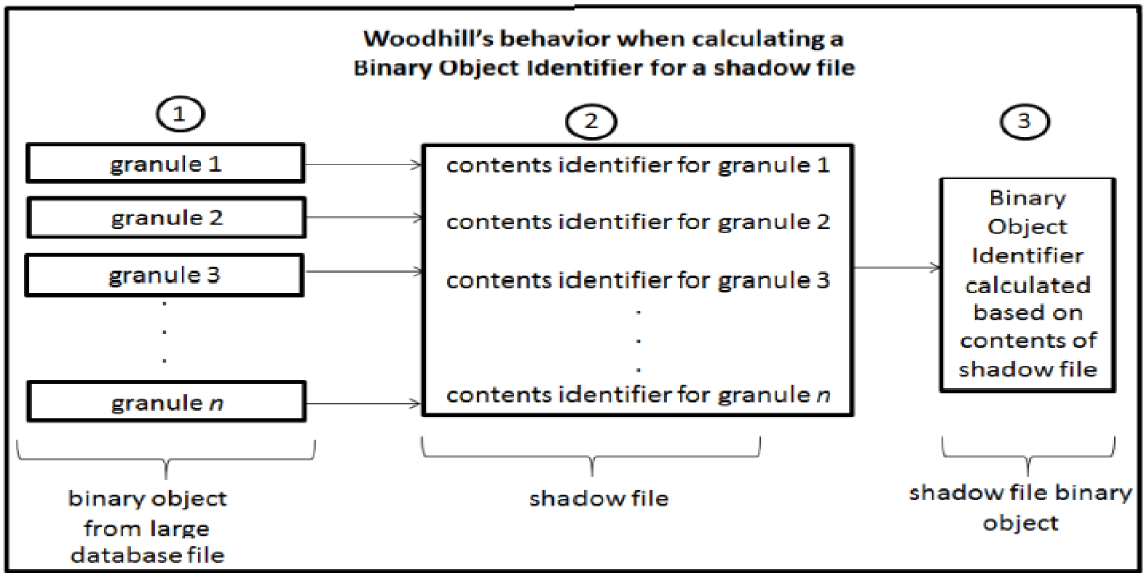
Shadow file identifiers are for the first and second data items

Claim 1 requires "ascertaining whether or not said first data item corresponds to said second data item based, at least in part, on said first value and said second value." In its petition, EMC takes the position that Woodhill meets this limitation because "by comparing binary objects of successive versions of shadow files, Woodhill by extension compares the binary objects underlying those shadow files." Pet. 56 (citing Ex. 1005, 9:5-28; Ex. 1009 ¶ 49). EMC further maintains that the comparison is "based, at least in part, on said first value" (the binary object identifier corresponding to a previous version of a shadow file) and "said second value" (the binary object identifier corresponding to the current version of the shadow file). *Id*. at 56-57 (citing Ex. 1005, 9:5-28; Ex. 1009 ¶ 49).

PersonalWeb counters that Woodhill's shadow file binary object identifiers are not "for the first data item" or "for the second data item." PO Resp. 11-15 (citing Ex. 2016 ¶¶ 36-40). According to PersonalWeb, "it would be highly unlikely, if not impossible, for a single 'shadow file' to be separated from a data stream to form a single standalone 'binary object,'" and that "the more likely scenario under this assumption would be that a 'binary object' would be made up of many shadow files." *Id*. at 11-12.

In its reply, EMC responds that "PersonalWeb's assumptions about Woodhill are directly contradictory to Woodhill's explicit disclosure." Reply 6 (citing Ex. 1005, 4:13-23; Ex. 1088 ¶¶ 14-15). We agree with EMC. Woodhill expressly discloses *dividing* files into *one or more* data streams, or *one or more* binary objects. Ex. 1005, 2:20-24 ("The present invention is further directed to a method for the management of storage space . . . storing data files comprised of one or more binary objects."); 4:22-23 (The DSM program "further divides each data stream into one or more binary objects."); 4:25-26 (A single binary object may represent a data stream.). Nothing in Woodhill suggests that *a plurality of shadow files* must be *combined* into a *single binary object*.

We also agree with EMC that a binary object identifier for a shadow file is "a hash of hashes" *for the underlying database binary object*. Reply 6-7. As Dr. Clark shows in his illustration (step 1), reproduced below, a binary object for a large database file (a first or second data item) is divided into a plurality of granules (a first or second plurality of parts) (Ex. 1088 ¶¶ 17-18; Ex. 1005, 14:53-15:16):


Woodhill's behavior when calculating a Binary Object Identifier for a shadow file
1
2
3
granule 1
granule 2
granule 3
granule n
contents identifier for granule 1
contents identifier for granule 2
contents identifier for granule 3
contents identifier for granule n
Binary Object Identifier calculated based on contents of shadow file
binary object from large database file
shadow file
shadow file binary object

As shown in step 2 of Dr. Clark's illustration (Ex. 1088 ¶ 17), Woodhill's DSM program calculates a contents identifier for each granule of the database binary object, using a hash function (first hash function), and stores each contents identifier in a shadow file (part value). Ex. 1088 ¶ 20 (citing Ex. 1005, 15:21-28). A binary object identifier (a first or second value) is calculated using a hash function (second hash function) based on the contents of the shadow file ("a hash of hashes"). *Id*. at ¶ 21 (citing Ex. 1005, 8:58-60; 15:21-28); *see also* Ex. 1005, 7:60-8:65. Therefore, the shadow file binary object identifier (step 3) is for the underlying database binary object (step 1). *Id*. We credit Dr. Clark's testimony as it is consistent with the explicit disclosure of Woodhill.

For the reasons stated above, EMC provides sufficient explanations and evidence to show that Woodhill describes obtaining a first value for the first data item and a second value for the second data item, as well as "ascertaining whether or not said first data item corresponds to said second data item based, at least in part, on said first value and said second value," as required by claim 1.

Conclusion

For the foregoing reasons, we hold that EMC has demonstrated by a preponderance of the evidence that claim 1 is anticipated by Woodhill.

D. Claim 1 – Anticipated by Kantor

EMC asserts that claim 1 is unpatentable under 35 U.S.C. § 102(b) as anticipated by Kantor. Pet. 28-36. In support of the asserted ground of unpatentability, EMC provides detailed explanations as to how each claim element, arranged as recited in the claim, is disclosed by Kantor. *Id*. EMC also directs our attention to the declaration of Dr. Clark. *Id.* (citing Ex. 1009 ¶¶ 3-4, 17-25).

In its patent owner response, PersonalWeb counters that Kantor does not describe "applying a first function comprising a hash to each of a plurality of parts of the first data item," as recited in claim 1. PO Resp. 15-25. PersonalWeb also alleges that Kantor is not a "printed publication" within the meaning of 35 U.S.C. § 102(b). *Id.* at 27-34. In support of its argument, PersonalWeb proffers Dr. Dewar's declaration (Ex. 2016 ¶¶ 43-55) and Mr. Todd Thompson's declaration (Ex. 2014).

Upon review of the parties' arguments and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claim 1 is unpatentable under 35 U.S.C. § 102(b) as being anticipated by Kantor. We also determine that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

<u>Kantor</u>

Kantor describes a method of identifying duplicate files. Ex. 1004, 2-4, 48-49. In particular, Kantor applies a hash function (*e.g.*, a cyclic residue check or cyclic redundancy check (CRC)) to each file within a zip file to obtain the contents signature for each file. *Id.* at 6-8, 48-49. Each contents signature is a string of bits generated from the contents of a file. *Id.*

For each zip file, Kantor creates zip-file contents signatures by hashing the contents signatures for the files contained within the zip file ("a hash of hashes"). *Id.* at 2, 9. As described by Kantor, this is done by "adding together all the 32_bit CRC's for the files in the zipfile, modulo

2^32, separately adding together their uncompressed file_lengths modulo 2^32, and then arranging the two resulting hexadecimal numbers as a single structure." *Id*. at 9. Dr. Clark testifies that addition modulo 2^32 is another well-known simple hashing function that uses addition to calculate a value for a file based on the file's contents. Ex. 1009 ¶ 20. Kantor further compares the zip-file contents signatures to check for duplicate files. Ex. 1004, 2 of Preface, 5, 9.

According to Kantor, contents signatures and zip-file contents signatures are useful to identifying files that have the same contents stored on the electronic bulletin board systems. Ex. 1004, 2 of Preface, 5, 9. For example, when uploading a zip file, the system determines whether that zip file already exists in the system using the zip-file contents signature, and determines whether the inner files of that zip file already exist in the system using the contents signatures for the inner files. *Id*. at 9.

Whether Kantor is a "printed publication"

In its petition, EMC takes the position that Kantor is a "printed publication" under 35 U.S.C. § 102(b). Pet. 28. EMC asserts that Kantor has been publicly available since August 1993, which is prior to the critical date, April 11, 1995, one year before the earliest priority date claimed by the '544 patent. *Id*. at 3. To substantiate its position, EMC explains that Kantor is "a published manual that describes a software program called the Frederick W. Kantor Contents-Signature System Version 1.22 ('FWKCS')." *Id*. at 28 (citing Ex. 1004, Title Page). EMC maintains that Dr. Frederick W. Kantor distributed Kantor—the user manual (version 1.22), the version relied upon by EMC (*see* Ex. 1004)—with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where Kantor could be downloaded by anyone. Pet. 3, n.1 (citing Ex. 1004, 3, 158-159). According to EMC, Kantor was accessible to others in the relevant community of the users and system operators of electronic Bulletin Board Systems. *Id.* As support, EMC proffers a declaration of Mr. Michael A. Sussell (Ex. 1049) and declarations of Mr. Jason S. Sadofsky (Exs. 1077, 1087).

In its patent owner response, PersonalWeb counters that Kantor is not a "printed publication." PO Resp. 27-34. In particular, PersonalWeb alleges that EMC has not established that the specific version of Kantor existed prior to the critical date. *Id.* at 29. PersonalWeb contends that there is no evidence that Kantor was disseminated publicly, catalogued, or indexed in a meaningful way. *Id.* at 32. It is PersonalWeb's view that EMC fails to establish that one with ordinary skill in the art, exercising reasonable diligence, would have located Kantor prior to the critical date. *Id.* at 30.

We have reviewed parties' arguments and supporting evidence. Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. Rather, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

The determination of whether a given reference qualifies as a prior art "printed publication" involves a case-by-case inquiry into the facts and circumstances surrounding the reference's disclosure to members of the public. *In re Klopfenstein*, 380 F.3d 1345, 1350 (Fed. Cir. 2004). The key inquiry is whether the reference was made "sufficiently accessible to the public interested in the art" before the critical date. *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989); *In re Wyer*, 655 F.2d 221, 226 (CCPA 1981). "A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it . . . ." *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006) (citation omitted).

Indexing is not "a necessary condition for a reference to be publicly accessible," but is only one among many factors that may bear on public accessibility. *In re Lister*, 583 F.3d 1307, 1312 (Fed. Cir. 2009). In that regard, "while often relevant to public accessibility, evidence of indexing is not an absolute prerequisite to establishing online references . . . as printed publications within the prior art." *Voter Verified, Inc. v. Premier Election Solutions, Inc.*, 698 F.3d 1374, 1380 (Fed. Cir. 2012).

Contrary to PersonalWeb's assertion that Kantor did not exist prior to the critical date and there is no evidence that Kantor was disseminated publicly, Kantor itself shows a copyright date of "1988-1993" and a posted date of "1993 August 10." Ex. 1004, Title Page, the first page after the Title Page ("All of the programs and documents, comprising the entire contents of this Authenticity Verification Zip file FWKCS122.ZIP, together with this

Zipfile itself, are, in accordance with their respective dates of creation or revision, (C) Copyright Frederick W. Kantor 1988-1993.”). Kantor also states:

> The FWKCS(TM) Contents_Signature System has become a robust platform for supporting contents_signature functions. FWKCS provides many functions and options for application in a public, commercial, school, institutional, or governmental environment. Extensive technical support is of special value in helping such users to benefit more fully from these many features.
>
> Registered FWKCS hobby BBS users are able to receive a modest amount of assistance, and are invited to participate in the FWKCS conference on The Invention Factory BBS, echoed via Execnet.
>
> Commercial, school, institutional, and governmental users, with their special support needs, are invited to discuss terms for obtaining such assistance.
>
> . . . .
>
> To get a new version of FWKCS, download FWKCSnnn.ZIP from The Invention Factory BBS, where nnn is the new version number without a decimal point. These special downloads are available at no fee, from a 43_line hunt_up group of USR Dual Standard modems, at 2400-16800 bits/sec (including V32.bis).

Ex. 1004, 158-159. It is clear from Kantor that, during the 1988-1993 timeframe, Dr. Kantor had posted many versions of his software and user manual—including Kantor (version 1.22), the version relied upon by EMC (Ex. 1004)—on electronic Bulletin Board Systems.

Mr. Sussell, the co-owner and system operator of the Invention Factory Bulletin Board System, testifies that the Invention Factory Bulletin

Board System is a computer system that allows users to share files, messages, and articles, as well as search, upload, and download files. Ex. 1049 ¶¶ 3-4. According to Mr. Sussell, he and his wife launched the Invention Factory Bulletin Board System in 1983, and it had over 3,000 subscribers by mid-1993. *Id.* ¶ 6. Mr. Sussell testifies that, by 1993, the system provided all users keyword search functionality and access to various descriptive and meaningful directories. *Id.* ¶¶ 8-10.

More importantly, Mr. Sussell testifies that the Invention Factory Bulletin Board System "extensively utilized and hosted current versions of FWKCS software on its [Bulletin Board System]" and "made publicly accessible and available the complete FWKCS ZIP file that contained both the software as well as related documentation such as user manuals" prior to the critical date. *Id.* ¶ 15; *see also id.* ¶¶ 16-27. Specifically, Mr. Sussell testifies that users would have found Kantor by performing keyword searches on the Invention Factory Bulletin Board System. *Id.* ¶ 21. Mr. Sussell also indicates that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about the software on the "Welcome" screen, and made the FWKCS Zip file available in four different directories. *Id.* ¶¶ 18-20. Mr. Sussell further testifies that computer disks that contain the FWKCS Zip file were distributed at various Bulletin Board System conferences. *Id.* ¶ 18.

Mr. Sadofsky, a technology archivist and software historian, testifies that he personally verified the authenticity of Kantor—the user manual (version 1.22), the version relied upon by EMC (Ex. 1004)—by comparing it with a "1993 archived" version, and determined that Kantor is identical to the "1993 archived" version. Ex. 1077 ¶¶ 14-17. Mr. Sadofsky testifies that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id*. ¶ 16; *see also* Ex. 1087 ¶¶ 10-11; Ex. 2014 ¶ 5. According to Mr. Sadofsky, Kantor was publicly accessible prior to the critical date. Ex. 1077 ¶¶ 13, 16-17.

PersonalWeb also asserts that Kantor was buried and hidden in the zip file in a manner such that "it would not have been located and accessed by persons interested and ordinarily skilled in the art exercising reasonable diligence even if they had access to the ZIP file." PO Resp. 33-34 (citing Ex. 2014). However, PersonalWeb's supporting evidence, Mr. Thompson's declaration (Ex. 2014), does not substantiate PersonalWeb's assertion. Upon review of Mr. Thompson's declaration, we observe that Mr. Thompson downloaded the FWKCS Zip file without any difficultly. Ex. 2014 ¶ 5. Significantly, Mr. Thompson did not follow the instructions provided with the zip file, nor did he use the appropriate computer environment (DOS 3.0 or an IBM OS/2 2.0) that was used normally in 1993-1994 timeframe. Ex. 2014 ¶¶ 6-11; Ex. 1087 ¶¶ 5, 14. Instead, he used non-compatible software (DOS 8.0 and 32-bit Windows XP operating system that was released in 2001). *Id*. Once he followed the instructions and unzipped the FWKCS Zip file, Mr. Thompson located Kantor without difficulty. Ex. 2014 ¶¶ 20-22.

Mr. Sadofsky confirms that the README.TXT file provides simple instructions and, if a user follows the instructions and uses the operating system that was used normally in 1993-1994 timeframe, the user could locate Kantor without difficulty. Ex. 1087 ¶¶ 13-17. In fact, Mr. Sadofsky demonstrated, in his declaration, several relatively easy ways for a user to access Kantor—with or without installing the software, and with or without help screens. Ex. 1087 ¶¶ 8-16 (II. README.TXT); ¶¶ 17-20 (III. GETLOOK.BAT); ¶¶ 21-22 (IV. FWKCS122 Start Screen and In-Program Help). Based on the evidence before us, we determine that Kantor was available to the extent that persons interested and ordinarily skilled in the art, exercising reasonable diligence, could locate it.

PersonalWeb's argument that EMC's witnesses personally did not post or review Kantor prior to the critical date also is unavailing. PO Resp. 29-31 (citing Ex. 2008, 52-55; Ex. 2013, 29-30; Ex. 2015, 98). It is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986) (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible"); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'"); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would know of the invention.).

The evidence on this record sufficiently supports that Kantor was posted on a publicly accessible site—the Invention Factory Bulletin Board System—well known to those interested in the art and could be downloaded and retrieved from that site, and therefore Kantor, an electronic publication, is considered a "printed publication" within the meaning of 35 U.S.C. § 102(b). *See Wyer*, 655 F.2d at 226 (An electronic publication, including an on-line database or Internet publication, is considered to be a "printed publication" "upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it and recognize and comprehend therefrom the essentials of the claimed invention without need of further research or experimentation.").

For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b). Therefore, EMC may rely upon Kantor for its asserted grounds of unpatentability under 35 U.S.C. §§ 102(b) and 103(a).

<u>Applying a hash function to each part of the zip file</u>

Claim 1 requires "for a first data item comprising a first plurality of parts, . . . applying a first function to each part of said first plurality of parts to obtain a corresponding part value for each part of said first plurality of parts . . . wherein said first function comprises a first hash function." Claim 1 also requires "obtaining a first value for the first data item, said first value obtained by applying a second function to the part values of said first plurality of parts of said first data item, said second function comprising a second hash function."

In its petition, EMC takes the position that Kantor describes the aforementioned limitations. EMC explains that Kantor discloses a "data item" (a zip file) comprising a "first plurality of parts" (the data files within the zip file). Pet. 34 (citing Ex. 1004, 2-3, 48-49; Ex. 1009 ¶ 19). Indeed, Kantor applies a CRC hash function (a first hash function) to the inner files of the zip file (the first plurality of parts) to obtain a contents signature for each inner file (part value). *Id*. (citing Ex. 1004, 48-49; Ex. 1009 ¶ 19). As to the "second hash" limitation, Dr. Clark testifies that Kantor discloses creating zip-file contents signatures for each zip file on the system by hashing the contents signatures for the individual files in the zip file ("a hash of hashes"). Ex. 1009 ¶ 20 (citing Ex. 1004, 9).

In its patent owner response, PersonalWeb counters that Kantor "teaches away" from applying a hash function to each of a plurality of parts of the first data item. PO Resp. 16. PersonalWeb also alleges that Kantor does not apply the CRC hash function to the parts of a zip file because the function is applied to *uncompressed* files before they are *compressed* and packaged into the zip file. *Id*. (citing Ex. 2016 ¶¶ 43-55). According to PersonalWeb, the CRC hash function is applied to different bits (uncompressed files) than the bits (compressed files) that make up the inner files in the zip file (the alleged data item), and therefore, the CRC hash function is not applied to the *compressed* inner files that are parts of the zip file in determining the zip-file contents signature of the zip file. *Id*. at 16-23.

At the outset, we note that, although a "teaching away" argument could be relevant to an obviousness analysis, "whether a reference teaches away from an invention is inapplicable to an anticipation analysis." *ClearValue, Inc. v. Pearl River Polymers, Inc.*, 668 F.3d 1340, 1344 (Fed. Cir. 2012) (citing *Celeritas Techs., Ltd. v. Rockwell Int'l Corp.*, 150 F.3d 1354, 1361 (Fed. Cir. 1998)) (quotation marks omitted).

In any event, we are not persuaded by PersonalWeb's arguments and expert testimony, as they rest on the erroneous premise that Kantor's data files contained within a zip file must be *compressed files*. Rather, we agree with EMC that "nothing in Kantor limits the 'inner files' of a zip file to *compressed* files" and Kantor's program works with zip files of all forms. Reply 7-8 (citing Ex. 1088 ¶¶ 26-28; Ex. 1084, 262-63; Ex. 1004, 2, 9). As Dr. Clark notes, PersonalWeb's evidence shows that zip files are not always compressed, as the standard zip-file format defines seven compression methods, including "Compression method 0," which does not compress the inner files when packaging them into a zip file. Ex. 1088 ¶ 26 (citing Ex. 2007, 3; Ex. 1084, 262).

Dr. Dewar's reliance on Kantor's statements regarding file compression ratio to support his testimony—"Kantor confirms that the inner files in the ZIP files described in Kantor are compressed"—is misplaced. Ex. 2016 ¶ 46 (citing Ex. 1004, 2 of Preface, 9, 55). The mere fact that

Kantor refers to a compression ratio does not support PersonalWeb's position that the inner files of a zip file must be *compressed*, because in the situation where "Compression method 0" is used, which does not compress the inner file, the file compression ratio is one. Contrary to Dr. Dewar's testimony, those portions of Kantor cited by Dr. Dewar do not require the inner files of a zip file to be compressed. Instead, the cited portions of Kantor merely state that the zip-file contents signature *depends on the contents of the files*, and provide examples of items that the zip-file contents signature do not depend upon. Ex. 1004, 2 of Preface ("FWKCS has the special ability to make a 'zipfile contents signature', ('zcs') which is *independent of* . . . the names and dates of files in the zipfile, zipped path information, and file compression ratio."); *id*. at 9 ("This has the desirable property that the resulting zcs *does not depend* on the names of the files, . . . nor on the method nor amount of compression . . ."); *id*. at 55 ("This zcs *does not depend* on the names, dates, compression ratios, order of appearance, zipped paths, nor comments, of files appearing in the zipfile, nor on the zipfile's name, date, nor zipfile comment.") (Emphases added).

We also agree with EMC that, even if Kantor only used compressed inner files, Kantor still would describe the disputed claim limitations, as the first function would *comprise a CRC hash function and a compression function*. Reply 8 (citing Ex. 1088 ¶ 28). Indeed, because claim 1 recites the open-ended phrase "comprising" when describing what the first function includes ("wherein said first function *comprises* a first hash function"), the first function is not limited to just a hash function. PersonalWeb does not explain adequately why the "first function" cannot comprise more than a hash function. Moreover, *compressing* a file merely changes *the format* of the file, but it does not change *the contents* of the file. In other words, both compressed and uncompressed versions of an inner file have the *same contents* (a corresponding sequence of bits). As discussed above, Kantor's contents signatures are generated based on *the contents* of the files using the CRC hash function (Ex. 1004, 6-8), and Kantor's zip-file contents signatures depend on *the contents* of the files and do not depend on the format of the files (Ex. 1004, 2 of Preface, 9, 55). Claim 1 does not place any limitation on *the format* of the plurality of parts ("wherein each part of said first plurality of parts *comprises* a corresponding sequence of bits"). Therefore, PersonalWeb's argument that the CRC hash function applies to uncompressed files before they are compressed and packaged into the zip file is unavailing.

We are not persuaded by PersonalWeb's argument that a zip file may include information in addition to the inner files of the zip file (e.g., headers) and, therefore, Kantor's CRC hash function does not apply to "each part of said first plurality of parts." That argument is not commensurate within the scope of claim 1. *See In re Self*, 671 F.2d 1344, 1348 (CCPA 1982) (It is well established that limitations not appearing in the claims cannot be relied upon for patentability.). Claim 1 recites "for a first data item *comprising* a first plurality of parts" (emphasis added). As discussed in the claim construction analysis above, the claim term "data item" includes a *portion* of a file. EMC relies on Kantor's disclosure of a zip file to describe "a first data item" and the data files within the zip file to describe "a first plurality of parts." Therefore, we agree with EMC (Reply 10, n.3) that "nothing in claim 1 prohibits the inclusion of [the] addition information in the 'data item,' (*comprising* a first plurality of parts)."

Additionally, we are not persuaded by PersonalWeb's argument that Kantor merely "reads" the CRC values from the zip file and the uncompressed file lengths (sizes) from the zip file, and does not apply a second hash function in determining the zip-file contents signatures. PO Resp. 21. According to Kantor, contents signatures are generated from the contents of the inner files of a zip file by applying a CRC hash function to the inner files. Ex. 1004, 48-49 ("Make a 'File contents signature' for (each) File in zipfile(s). . . . The output includes the contents_signature for the file inside the zipfile (using the 32_bit CRC and the uncompressed length of that file). . ."); *id*. at 1-2 of Preface, 6, 9, 48-49, 55. PersonalWeb also narrowly focuses on Kantor's reading steps and ignores Kantor's other steps for determining a zip-file contents signature—"adding together all the 32_bit CRC's for the files in the zip file, modulo 2^32, separately adding together their uncompressed_file_lengths modulo 2^32, and then arranging the two resulting hexadecimal numbers as a single structure." Ex. 1004, 9.
Dr. Clark testifies that addition modulo 2^32 is a well-known simple hashing function that uses addition to calculate a value for a file based on the contents of the file. Ex. 1009 ¶ 20 (citing Ex. 1011). Dr. Clark's testimony is consistent with Kantor's disclosure that the resulting zip-file contents signature "does not depend on the names of the files, the dates of the files,

the order in which they appear in the zip file, nor on the method nor amount of compression, nor does it depend on comments." Ex. 1004, 9.

Given the express disclosure of Kantor, we determine that EMC has demonstrated sufficiently that Kantor's CRC hash function (a first hash function) applies to "each part of said first plurality of parts" of the first data item, as recited in claim 1.

E. Claim 1 – Obvious over Kantor and Woodhill

EMC asserts that claim 1 is unpatentable under 35 U.S.C. § 103(a) as obvious over Kantor in view of Woodhill. Pet. 36. In particular, EMC submits that "in the event PersonalWeb contends that Kantor does not satisfy the claim limitation of a 'plurality of parts' of a data item, a person of ordinary skill would have found it obvious to modify Kantor to meet that limitation." *Id.* EMC maintains that dividing a file into parts (e.g., dividing a file into a plurality of binary objects or granules) was a well-known technique to handle large files, as evidence by Woodhill, to reduce the amount of data that must be transmitted. *Id.* (citing Ex. 1005, 4:14-30; 14:52-15:8; Ex. 1009 ¶ 26).

In its patent owner response, PersonalWeb counters that the obviousness ground of unpatentability does not cure the deficiencies of Kantor. PO Resp. 25. PersonalWeb essentially relies upon the same arguments presented above with respect to the anticipation ground of unpatentability based on Kantor. *Id.* at 25-26. As discussed above, we have addressed those arguments and determined that they are unavailing.

PersonalWeb also alleges that one with ordinary skill in the art would not have modified Kantor to include small data items, because "there is no need for this in Kantor as Kantor is concerned with avoiding duplicate files and not with creating duplicates by backing up files." *Id*. at 26. However, PersonalWeb's argument improperly focuses on Woodhill's *entire back-up procedure*. EMC's proposed modification does not require incorporating Woodhill's *entire back-up procedure* into Kantor's method of identifying duplicate files. In fact, EMC merely relies upon Woodhill's technique of *dividing a file into a plurality of parts*. Pet. 36. "It is well-established that a determination of obviousness based on teachings from multiple references does not require an actual, physical substitution of elements." *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); *see also In re Etter*, 756 F.2d 852, 859 (Fed. Cir. 1985) (en banc) ("[T]he criterion [is] not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the prior art as a whole."). "To justify combining reference teachings in support of a [ground of unpatentability] it is not necessary that a device shown in one reference can be physically inserted into the device of the other." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).

Moreover, incorporating Woodhill's technique of dividing a file into a plurality of parts (Ex. 1005, 4:14-30; 14:52-15:8; Ex. 1009 ¶ 26) into Kantor's method of identifying duplicate files would not have been beyond the level of an ordinarily skilled artisan. *KSR*, 550 U.S. at 417 ("[I]f a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.”).

PersonalWeb further submits that its evidence of non-obviousness outweighs EMC’s evidence of obviousness. PO Resp. 26-27. In support of its argument, PersonalWeb directs our attention to three licensing agreements, as well as the declaration of Mr. Kevin Bermeister. *Id*. at 27 (citing Exs. 2010-12; Ex. 2009 ¶¶ 3-9). PersonalWeb argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between claim 1 of the ’544 patent and the above-identified license agreements. Reply 12-13. EMC argues that each of the licenses granted rights to more than just claim 1, and involved related parties with interlocking ownership and business interests. *Id*. We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and claim 1.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *GPAC*, 57 F.3d at 1580. Further, without a showing of nexus, “the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness” when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000); *Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus. Neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010, 2011, 2012) are directed to the claimed subject matter recited in claim 1. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in claim 1. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Furthermore, even if we assume that above-identified licenses establish some degree of industry respect for the claimed subject matter recited in claim 1, that success is outweighed by the strong evidence of obviousness over Kantor and Woodhill discussed above.

Based on the record before us, including the evidence of non-obviousness presented by PersonalWeb and the evidence of obviousness presented by EMC, we conclude that EMC has demonstrated by a preponderance of the evidence that claim 1 would have been obvious over the combination of Kantor and Woodhill.

F. EMC's Motion to Exclude

EMC seeks to exclude the following exhibits: (1) three license agreements (Exs. 2010-12); (2) Mr. Bermeister's declarations (Exs. 2009, 2017) relating to those license agreements; and (3) Mr. Thompson's declaration (Ex. 2014). Paper 50 ("Pet. Mot."). PersonalWeb filed the license agreements and Mr. Bermeister's declarations as evidence of non-obviousness to rebut EMC's assertion that claim 1 would have been obvious over the combination of Kantor and Woodhill. PO Resp. 12-13. As to Mr. Thompson's declaration, PersonalWeb proffered that evidence to support its assertion that Kantor—a user manual that was disseminated publicly with the software in a zip file—was not made sufficiently accessible to a person interested and ordinarily skilled in the art. *Id*. at 32-34.

With respect to the license agreements and Mr. Bermeister's declarations (Exs. 2010-2012; Exs. 2009, 2017), EMC argues that they are irrelevant under Federal Rule of Evidence 402[4], highly prejudicial, confusing, and misleading under Federal Rule of Evidence 403. Pet. Mot. 1-13. As to Mr. Thompson's declaration, EMC argues that it should be excluded under Federal Rule of Evidence 402. *Id*. at 14-15. In particular, EMC alleges that: (1) Mr. Thompson does not possess the skill of a person within ordinary skill in the art (*id*. at 14, citing Ex. 1082, 13-14);
(2) Mr. Thompson did not use compatible software from the relevant time period (*id*. at 14, citing Ex. 1082, 40-41; Ex. 2014, 4, 6); and
(3) Mr. Thompson did not follow the instructions provided with the zip file (*id*. at 14, citing Ex. 1082, 32-35).

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding

[4] As stated in 37 C.F.R. § 42.62, the Federal Rules of Evidence generally apply to proceedings, including *inter partes* reviews.

PersonalWeb's supporting evidence, we have determined that Kantor is a "printed publication" under 35 U.S.C. § 102(b), and EMC has demonstrated by a preponderance of the evidence that claim 1 is unpatentable over the combination of Kantor and Woodhill.

Accordingly, EMC's motion to exclude evidence is *dismissed* as moot.

G. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following items of evidence: (1) Kantor (Ex. 1004); (2) certain documents that corroborating witnesses' knowledge and recollections (Exs. 1046-1048, 1051-1054, 1073, 1074, 1079-1081) and the portions of witnesses' testimony regarding these documents; (3) the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66); and (4) Dr. Clark's rebuttal declaration (Ex. 1088 ¶¶ 26-27, 30). Paper 48 ("PO Mot.").

EMC opposes PersonalWeb's motion to exclude. Paper 55 ("Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 58 ("PO Reply"). For the reasons stated below, PersonalWeb's motion to exclude is *denied*.

1. Kantor

PersonalWeb alleges that Kantor should be excluded as unauthenticated and inadmissible hearsay under Federal Rules of Evidence 901 and 902. PO Mot. 1, 6. In particular, PersonalWeb argues that "[n]o witness of record has personal knowledge of Kantor existing prior to [the critical date], and electronic data such as Kantor is inherently untrustworthy because it can be manipulated from virtually any location at any time." *Id*. at 2-4. According to PersonalWeb, the dates provided by Kantor are inadmissible hearsay because Kantor is not self-authenticating. *Id*. at 2, 5-6.

EMC argues that Kantor has been authenticated under Federal Rules of Evidence 901, and that the document is not hearsay, because it is being offered for what it describes—not for the truth of its disclosures. Opp. 1-8. In particular, EMC disagrees with PersonalWeb that Kantor cannot be authenticated without direct testimony from a witness with personal knowledge that Kantor existed prior to the critical date. Opp. 1. EMC asserts that it need "only produce evidence 'sufficient to support a finding' that the reference 'is what the proponent claims it is.'" *Id*. at 1-2 (citing Fed. R. Evid. 901(a)). EMC also contends that testimony from Messrs. Sussell and Sadofsky provides sufficient evidence to authenticate Kantor. Opp. 1-6 (citing Exs. 1049, 1077, 1087).

In its reply, PersonalWeb argues that Federal Rules of Evidence identified by EMC are not applicable to Kantor, because Mr. Sussell did not post or review Kantor prior to critical date. PO Reply 1-5 (citing Ex. 2008, 52-55, 65). PersonalWeb also alleges that Kantor's authenticity is suspicious, as electronic data are inherently untrustworthy and there is no chain of custody. *Id*.

We have considered PersonalWeb's arguments as well as EMC's contentions and supporting evidence. We are not persuaded that Kantor should be excluded. At the outset, we disagree with PersonalWeb's position that a witness cannot authenticate a document, unless the witness is the author of the document or the witness has reviewed the document prior to the critical date. Federal Rule of Evidence 901(a) states that the authentication requirement is satisfied if the proponent presents "evidence sufficient to support a finding that the item is what the proponent claims it is." Therefore, neither a declaration from the author, nor evidence of someone actually viewing the document *prior to critical date,* is required to support a finding that the document is what it claims to be. *See Hall*, 781 F.2d at 899 (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible."); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'").

Further, it is well settled that an uninterrupted chain of custody is not a prerequisite to admissibility, but rather gaps in the chain go to weight of the evidence. *U.S. v. Wheeler*, 800 F.2d 100, 106 (7th Cir. 1986); *see also U.S. v. Aviles*, 623 F.2d 1192, 1198 (7th Cir. 1980) ("If the trial judge is satisfied that in reasonable probability the evidence has not been altered in any material respect, he may permit its introduction." (citation omitted)). There is a strong public policy for making all information filed in a quasi-judicial administrative proceeding available to the public, especially in an *inter partes* review, which determines the patentability of a claim in an issued patent. It is within the Board's discretion to assign the appropriate weight to be accorded to evidence.

Although Messrs. Sussell and Sadofsky, prior to the critical date, personally did not post or review the particular version of Kantor relied upon by EMC (Ex. 1004), they nevertheless have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. *See Hall*, 781 F.2d at 899; *Wyer*, 655 F.2d at 226; *Bayer*, 568 F.2d at 1361. Notably, Mr. Sussell, the co-founder and system operator of the Invention Factory Bulletin Board System, testifies that Dr. Kantor released the first version of his software on the Invention Factory Bulletin Board System in the 1980s, and the system continuously utilized and hosted current versions of the software and user manuals. Ex. 1049 ¶¶ 3, 13, 15. Mr. Sussell also testifies that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about Dr. Kantor's software on the "Welcome" screen, and made FWKCS Zip file—a zip file that contains both the software and user manual, Kantor—publicly accessible and available under four different directories. *Id.* ¶ 18. According to Mr. Sussell, the Invention Factory Bulletin Board System had over 3,000 subscribers, in the 1993 timeframe, and all of the users had the capability to perform keyword searches to retrieve FWKCS Zip file. *Id.* ¶¶ 6, 21.

Although we are cognizant that electronic documents generally are not self-authenticating, it has been recognized that "[t]o authenticate printouts from a website, the party proffering the evidence must produce some statement or affidavit from someone with knowledge of the website . . . for example a web master or someone else with personal knowledge would be sufficient." *St. Luke's Cataract and Laser Institute v. Sanderson*, 2006 WL 1320242, *2 (M.D. Fla. 2006) (quoting *In re Homestore.com, Inc. Sec. Litig.*, 347 F. Supp. 2d 769, 782 (C.D. Cal. 2004)) (internal quotation marks omitted); Ex. 2024; *see also Market-Alerts Pty. Ltd. v. Bloomberg Finance L.P.*, 922 F. Supp. 2d 486, 493, n.12 (D. Del. 2013) (citing *Keystone Retaining Wall Sys., Inc. v. Basalite Concrete Prods., LLC*, 2011 WL 6436210, *9 n.9 (D.Minn. 2011)) (Documents generated by a website called the Wayback Machine have been accepted generally as evidence of prior art in the patent context.); *U.S. v. Bansal*, 663 F.3d 634, 667-68 (3d. Cir. 2011) (concluding that the screenshot images from the Internet Archive were authenticated sufficiently under Federal Rule of Evidence 901(b)(1) by a witness with personal knowledge of its contents, verifying that the screenshot the party seeks to admit are true and accurate copies of Internet Archive's records).

Here, Mr. Sadofsky, who is a technology archivist and software historian and currently is an archivist for the Internet Archive, testifies that he launched the website textfiles.com and a subdomain cd.textfiles.com to collect software, data files, and related materials from Bulletin Board Systems. Ex. 1077 ¶¶ 9-11. According to Mr. Sadofsky, textfiles.com and cd.textfiles.com are dedicated to preserving, archiving, and providing free access to unaltered historical software programs and information that initially were made available on the Bulletin Board Systems. *Id*.

Mr. Sadofsky states that he previously archived the FWKCS Zip file (FWKCS122.ZIP) that contains Dr. Kantor's software and user manual to cd.textfiles.com from his own copy of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM. *Id.* ¶ 14 (citing Ex. 1048).
Mr. Sadofsky also testifies that he personally verified the authenticity of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—by comparing it with the "1993 archived" version and determined that Kantor is identical to the "1993 archived" version. Ex. 1077 ¶¶ 13-15. Mr. Sadofsky confirms that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id.* ¶ 16; *see also* Ex. 1087 ¶¶ 10-11; Ex. 2014 ¶ 5. Mr. Sadofsky concludes that Kantor was publicly accessible prior to the critical date. Ex. 1077 ¶¶ 13, 16. Therefore, we agree with EMC that Kantor has been authenticated sufficiently to warrant its admissibility under Federal Rules of Evidence 901(b)(1), (b)(3), and (b)(4).

In addition, we agree with EMC that Kantor also has been authenticated as an "ancient document" under Federal Rule of Evidence 901(b)(8).[5] Opp. 7. Kantor is "at least 20 years old and can be found in . . . an October 1993 *Simtel* CD-ROM – a place where an authentic 20-year old document distributed through a [Bulletin Board System] would likely be."

[5] Fed. R. Evid. 901(b)(8). Evidence About Ancient Documents or Data Compilations. For a document or data compilation, evidence that it:
(A) is in a condition that creates no suspicion about its authenticity;
(B) was in a place where, if authentic, it would likely be; and
(C) is at least 20 years old when offered.

Id.; Ex. 1077 ¶¶ 7-8; *see also* Fed. R. Evid. 901(b)(8) 2012 Adv. Comm. Note ("The familiar ancient document rule of the common law is extended to include data stored electronically or by other similar means."). Moreover, testimony of Messrs. Sussell and Sadofsky has established sufficiently that Kantor is in a condition that creates no suspicion about its authenticity. Exs. 1049, 1077, 1087.

PersonalWeb does not present sufficient or credible evidence to the contrary. Based on the evidence before us, we determine that Kantor has been authenticated sufficiently under Federal Rules of Evidence 901(b)(1), (b)(3), (b)(4), and (b)(8) to warrant its admissibility.

PersonalWeb's hearsay argument regarding Kantor also is unavailing. As EMC notes (Opp. 8), a "prior art document submitted as a 'printed publication' under 35 U.S.C. § 102(a) is offered simply as evidence of what it described, not for proving the truth of the matters addressed in the document." *See, e.g., Joy Techs., Inc. v. Manbeck*, 751 F. Supp. 225, 233 n.2 (D.D.C. 1990), *judgment aff'd*, 959 F.2d 226 (Fed. Cir. 1992); Fed. R. Evid. 801(c) 1997 Adv. Comm. Note ("If the significance of an offered statement lies solely in the fact that it was made, no issue is raised as to the truth of anything asserted, and the statement is not hearsay."). Therefore, Kantor is not hearsay under Federal Rule of Evidence 801(c).

We further agree with EMC that the posted date of "1993 August 10" or the copyright date of "1988-1993" on the Title page of Kantor is not a basis for excluding Kantor, as testimony from Messrs. Sussell and Sadofsky sufficiently establishes that Kantor existed as of August 10, 1993, prior to the critical date. Opp. 8. More importantly, the computer-generated timestamp—August 10, 1993, at 1:22 AM—of the "1993 archived" version of Kantor (Ex. 1077 ¶¶ 14-15; Ex. 1087 ¶¶ 10-11; Ex. 2014 ¶ 5) also independently corroborates Kantor's existence as of August 10, 1993. *See, e.g.*, *U.S. v. Khorozian*, 333 F.3d 498, 506 (Fed. Cir. 2003) (concluding that an automatically generated time stamp on a fax was not a hearsay statement because it was not uttered by a person). Accordingly we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Kantor, as impermissible hearsay.

For the foregoing reasons, we decline to exclude Kantor.

2. Documents Corroborating Witnesses' Knowledge and Recollections

PersonalWeb asserts that certain documents submitted by EMC (Exs. 1046-1048, 1051-1054, 1073, 1074, 1079-1081) and the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) regarding these documents should be excluded because the documents have not been authenticated properly and are inadmissible hearsay. PO Mot. 6-9. PersonalWeb argues that EMC "has not established that any of these documents existed prior to the critical date, and no witness has personal knowledge of their alleged existence prior to April 11, 1995." *Id*. at 7. PersonalWeb further maintains that the documents that are Exhibits 1052, 1053, 1073, and 1074 are irrelevant, prejudicial, and confusing, as they discuss a version of Kantor different than the version relied upon by EMC (version 1.22, Ex. 1004). *Id*. at 8-9.

EMC responds that its witnesses provided those documents to corroborate their independent knowledge and recollections. Opp. 10. EMC asserts that the documents have been authenticated under Federal Rules of Evidence 901-902 and fall within a hearsay exception under Federal Rules of Evidence 803-807. *Id*. at 10-12. We are persuaded by EMC's arguments.

As the movant, PersonalWeb has the burden of proof to establish that it is entitled to the requested relief. 37 C.F.R. § 42.20(c). As discussed previously, we disagree with PersonalWeb that documents cannot be authenticated without direct testimony from the author or a witness who actually reviewed the documents prior to the critical date. *See* Fed. R. Evid. 901(a). Significantly, PersonalWeb's motion does not contain sufficient explanations why each document should be excluded. For instance, PersonalWeb does not explain adequately why the declaration of Mr. Sussell (Ex. 1049 ¶¶ 6, 8, 18, 27) is not sufficient to authenticate Exhibits 1051-1054, 1073, and 1074, or why the declarations of Mr. Sadofsky (Ex. 1077 ¶¶ 7-17; Ex. 1087 ¶¶ 10-16) are not sufficient to authenticate Exhibits 1046-48 and 1079-1081. *See* Fed. R. Evid. 901(b)(1).[6] Nor does PersonalWeb explain sufficiently why the following documents are not self-authenticated: (1) Exhibits 1046-1048 and 1051 – articles that have LexisNexis® trade inscriptions; (2) Exhibits 1073 and 1074 – Usenet newsgroup periodicals that have Usenet trade inscriptions; and (3) Exhibit

[6] Fed. R. Evid. 901(b)(1). Testimony of a Witness with Knowledge. Testimony that an item is what it is claimed to be.

1048 – a photograph of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM, that has Simtel trade inscriptions. *See* Fed. R. Evid. 902(6)-(7).[7]

In its motion, PersonalWeb fails to identify, specifically, the textual portions of the aforementioned exhibits that allegedly are being offered for the truth of the matter asserted, yet seeks to exclude the entirety of each exhibit. The burden should not be placed on the Board to sort through the entirety of each exhibit and determine which portion of the exhibit PersonalWeb believes to be hearsay. Rather, PersonalWeb should have identified, in its motion, the specific portions of the evidence and provided sufficient explanations as to why they constitute hearsay. Furthermore, PersonalWeb does not explain adequately why the declarations of Messrs. Sussell and Sadofsky do not provide the proper foundation and corroboration for the exhibits.

To the extent PersonalWeb relies upon the same arguments with respect to Kantor for excluding the aforementioned exhibits, we have addressed those arguments above and determined that they are unavailing. We also

[7] Fed. R. Evid. 902. Evidence that Is Self-Authenticating
The following items of evidence are self-authenticating; they require no extrinsic evidence of authenticity in order to be admitted:

> . . . .
>
> (6) Newspapers and Periodicals. Printed material purporting to be a newspaper or periodical.
>
> (7) Trade Inscriptions and the Like. An inscription, sign, tag, or label purporting to have been affixed in the course of business and indicating origin, ownership, or control.

agree with EMC that the exhibits concerning prior versions of Kantor are relevant, and not prejudicial or confusing as alleged by PersonalWeb, because such circumstantial evidence provides context and corroboration for the witnesses' independent knowledge and recollection.

Furthermore, we are not persuaded that the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) should be excluded. As we discuss above and below in the next section, they have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. The documents they cite serve to corroborate their independent knowledge and recollection.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude Exhibits 1046-1048, 1051-1054, 1073, 1074, and 1079-1081, as well as the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) concerning those exhibits.

3. Declarations of Messrs. Sussell and Sadofsky

PersonalWeb argues that the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) should be excluded as hearsay under Federal Rule of Evidence 801 and inadmissible under Federal Rules of Evidence 802-807 for lack of foundation and personal knowledge, and Federal Rule of Evidence 702 as improper testimony, because the witnesses personally did not review Kantor (Ex. 1004) and Simtel (Ex. 1048) prior to the critical date. PO Mot. 9. PersonalWeb also argues that Messrs. Sussell and Sadofsky "are not qualified experts in the field." *Id*. at 11. PersonalWeb further alleges that Mr. Sadofsky's deposition (Ex. 2013, 30,

66) should be excluded, as it was responsive to a leading question and non-responsive to the question. *Id*.

EMC responds that the testimony of Messrs. Sussell and Sadofsky should not be excluded, because their testimony is based on their own personal knowledge and recollection, and the documents they cite serve to corroborate their independent knowledge and recollection. Opp. 13. EMC further explains that the witnesses have described thoroughly the underlying facts, and, therefore, the testimony should be admitted as relevant under Federal Rules of Evidence 401-402, supported by personal knowledge and foundation under Federal Rule of Evidence 602, and proper opinion testimony under Federal Rules of Evidence 701-703. *Id*. We find EMC's contentions have merit.

PersonalWeb's arguments rest on the erroneous premise that EMC's witnesses must have reviewed Kantor or Simtel personally prior to the critical date in order to provide competent testimony regarding Kantor or Simtel. As discussed previously, it is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See, e.g., Wyer*, 655 F.2d at 226.

Although Messrs. Sussell and Sadofsky are not experts related to the claimed subject matter of the '544 patent, each witness nevertheless has sufficient personal knowledge and working experience to provide competent testimony. *See Hall*, 781 F.2d at 899. Mr. Sussell was the co-owner and system operator of the Invention Factory Bulletin Board System from 1983 to 1996. Ex. 1049 ¶ 3. Mr. Sussell's testimony is based on his personal knowledge of the relevant facts related to the Invention Factory Bulletin Board System and Kantor. *Id*. at ¶ 2. Notably, Dr. Kantor specifically thanked Mr. Sussell in his user manual for hosting Dr. Kantor's software FWKCS and for Mr. Sussell's role in its development. Ex. 1004, 3 ("To Michael Sussell, sysop of The Invention Factory (R), home board for the support of FWKCS, for bringing the problem of duplicate files to my attention and for his help in testing . . . ."); *id.* at 6 ("When Michael Sussell, sysop of The Invention Factory (R) in New York, brought to my attention the problem of duplicate files with different names, these concepts provided valuable insight into how one might proceed.").

Mr. Sadofsky is a technology archivist and software historian, and, currently, works "for the Internet Archive, a non-profit digital library offering free universal access to books, movies, and music, as well as 342 billion archived webpages available through the Wayback Machine service." Ex. 1077 ¶ 3. Mr. Sadofsky also "directed the film, *The BBS Documentary,* an eight-episode documentary about the subculture born from the creation of the [Bulletin Board System]." *Id*. at ¶ 4. Mr. Sadofsky's testimony is based on his personal knowledge of the relevant facts related to Kantor and the "1993 archived" version of Kantor. *Id*. at ¶ 2; Ex. 1087 ¶ 2. For example, Mr. Sadofsky personally verified the authenticity of Kantor by comparing it with the "1993 archived" version, and determined that Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—is identical to the "1993 archived" version. Ex. 1077 ¶¶ 14-15.

Upon review of the evidence on the record, we agree with EMC that both Messrs. Sussell and Sadofsky have disclosed sufficient underlying facts to support their testimony. For instance, the computer-generated timestamp—August 10, 1993, 1:22 AM—associated with the "1993 archived" version of Kantor corroborates their testimony regarding Kantor's existence as of August 10, 1993. Ex. 1077 ¶¶ 14-15; Ex.1087 ¶¶ 10-11; Ex. 2014 ¶ 5.

As to Mr. Sadofsky's deposition (Ex. 2013, 30, 66), PersonalWeb does not explain sufficiently why that testimony should be excluded. PO Mot. 11. Moreover, Mr. Sadofsky's deposition (Ex. 2013, 30, 66) is consistent with his direct testimony (Ex. 1077 ¶¶ 14-16), and, therefore, it would not prejudice PersonalWeb even if such evidence is not excluded.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude the declarations of Messrs. Sussell and Sadofsky (Exs. 1049, 1077, 1087) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66).

4. Clark's Rebuttal Declaration

PersonalWeb alleges that Dr. Clark's rebuttal declaration (Ex. 1088 ¶¶ 26-27, 30) should be excluded, because it is irrelevant, prejudicial, confusing, and beyond the scope of this proceeding. PO Mot. 11-14. In particular, PersonalWeb argues that what Kantor's software allegedly could do is irrelevant because an *inter partes* review is limited to printed publications and patents. *Id*. at 11-12. PersonalWeb also contends that Dr. Clark's rebuttal declaration (Ex. 1088 ¶¶ 26-27) contradicts his prior deposition (Ex. 2015, 55, 59, 66-67) and constitutes new evidence that should have been presented earlier. *Id.* at 12-14.

EMC counters that Dr. Clark's rebuttal declaration (Ex. 1088 ¶¶ 26-27, 30) regarding what Kantor's software "could" do is relevant and admissible. Opp. 13-14. According to EMC, Dr. Clark's rebuttal declaration (Ex. 1088 ¶ 30) was submitted in response to PersonalWeb's argument that "Kantor's software reads part identifiers (CRC values) from a zip file instead of generating them" (PO Resp. 20-22; Ex. 2016 ¶¶ 49-50). *Id.* EMC points out that Dr. Clark's rebuttal declaration merely explains "what a person of skill in the art, reading the Kantor reference, would [have understood] the reference to disclose about Kantor's software, including the software's disclosed capabilities to generate CRC values when it cannot look them up." *Id.* at 14 (citing Ex. 1088 ¶ 30).

Having reviewed PersonalWeb's patent owner response and Dr. Clark's rebuttal declaration, we determine that Dr. Clark's testimony is reasonable rebuttal evidence in light of PersonalWeb's arguments submitted in its patent owner response. Notably, Dr. Clark's rebuttal declaration responds appropriately to the issue raised by PersonalWeb (PO Resp. 20-22; Ex. 2016 ¶¶ 49-50)—whether Kantor, a software user manual, describes generating a hash of hashes, as required by claim 1, when Kantor describes how the software calculates a zip-file contents signature. PersonalWeb has not demonstrated sufficiently that Dr. Clark's rebuttal testimony is irrelevant or exceeds the proper scope of reply evidence.

We also are not persuaded by PersonalWeb's argument that Dr. Clark's rebuttal declaration (Ex. 1088 ¶¶ 26-27) contradicts his earlier testimony (Ex. 2015, 55, 59, 66-67). Rather, we agree with EMC (Opp. 14-15) that Dr. Clark's rebuttal testimony that "zip files are not *always* compressed" (Ex. 1088 ¶¶ 26-27) is consistent with his earlier testimony that the inner files of a zip file are compressed *typically* (Ex. 2015, 55, 59, 66-67). Moreover, Dr. Clark's testimony (Ex. 1088 ¶¶ 26-27) is reasonable rebuttal evidence in light of the evidence submitted by PersonalWeb in support of its patent owner response. Dr. Clark merely points out in his rebuttal declaration that PersonalWeb's evidence also shows that zip files are not *always* compressed. Ex. 1088 ¶ 26 (citing Ex. 2007, 3 (The zip file format defines seven compression methods, including "Compression method 0," which does not compress the file.); Ex. 1084, 262 (Dr. Dewar agrees that "the zip file standard allows for uncompressed files.")).

For the foregoing reasons, we decline to exclude Dr. Clark's rebuttal declaration (Ex. 1088).

III. CONCLUSION

EMC has met its burden of proof by a preponderance of the evidence in showing that claim 1 the '544 patent is unpatentable based on the following grounds of unpatentability:

Claim	Basis	References
1	§ 102(e)	Woodhill
1	§ 102(b)	Kantor
1	§ 103(a)	Kantor and Woodhill

IV. ORDER

In consideration of the foregoing, it is

ORDERED that claim 1 of the ’544 patent is held unpatentable;

FURTHER ORDERED that EMC’s Motion to Exclude Evidence is *dismissed*;

FURTHER ORDERED that PersonalWeb’s Motion to Exclude Evidence is *denied*; and

FURTHER ORDERED that because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Peter M. Dichiara, Esq.
David L. Cavanaugh, Esq.
WILMER CUTLER PICKERING HALE & DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com

PATENT OWNER:

Joseph A. Rhoa, Esq.
Updeep S. Gill, Esq.
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

Trials@uspto.gov
571-272-7822

Paper 73
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION,
Petitioner,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00085
Patent 7,945,539 B2

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

CHANG, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

EMC Corporation ("EMC") filed a petition on December 16, 2012, requesting an *inter partes* review of claims 10, 21, and 34 of U.S. Patent No. 7,945,539 B2 ("the '539 patent"). Paper 5 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") filed a patent owner preliminary response. Paper 11 ("Prelim. Resp."). Taking into account the patent owner preliminary response, the Board determined that the information presented in the petition demonstrated that there was a reasonable likelihood that EMC would prevail with respect to at least one claim. Pursuant to 35 U.S.C. § 314, the Board instituted this trial as to claims 10, 21, and 34 of the '539 patent. Paper 18 ("Dec.").

After institution, PersonalWeb filed a patent owner response (Paper 40 ("PO Resp.")), and EMC filed a reply to the patent owner response (Paper 48 ("Reply")). Oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is entered pursuant to 35 U.S.C. § 318(a). We hold that claims 10, 21, and 34 of the '539 patent are unpatentable under 35 U.S.C. §§ 102 and 103.

[1] This proceeding, as well as IPR2013-00082, IPR2013-00083, IPR2013-00084, IPR2013-00086, and IPR2013-00087, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 72.

A. Related Proceeding

EMC indicates that the ’539 patent is the subject of litigation titled *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED (E.D. Tex.). Pet. 1.

B. The ’539 patent

The ’539 patent relates to a method for identifying a data item (e.g., a data file or record) in a data processing system, by using an identifier that depends on all of the data in the data item and only on the data in the data item. Ex. 1001, 1:45-48; 3:52-56. Thus, the identity of a data item is said to be independent of its name, origin, location, and address. *Id*. at 3:55-58. According to the ’539 patent, the system provides transparent access to any data item by reference only to its identity and independent of its present location. *Id*. at 4:11-13. Figure 10(b) of the ’539 patent, reproduced below, is a flow chart for determining an identifier of a data item.

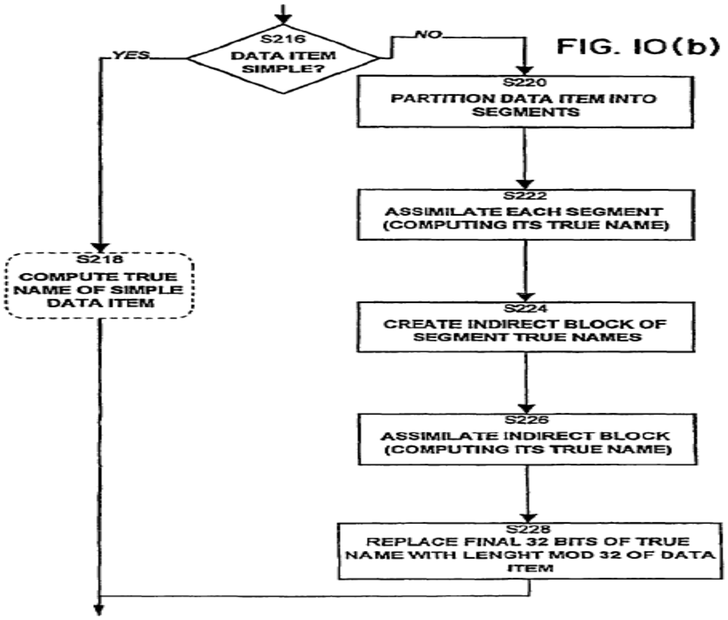

S216
DATA ITEM SIMPLE?
YES
NO
S218
COMPUTE TRUE NAME OF SIMPLE DATA ITEM
S220
PARTITION DATA ITEM INTO SEGMENTS
S222
ASSIMILATE EACH SEGMENT (COMPUTING ITS TRUE NAME)
S224
CREATE INDIRECT BLOCK OF SEGMENT TRUE NAMES
S226
ASSIMILATE INDIRECT BLOCK (COMPUTING ITS TRUE NAME)
S228
REPLACE FINAL 32 BITS OF TRUE NAME WITH LENGHT MOD 32 OF DATA ITEM

FIG. IO(b)

As shown in Figure 10(b) of the '539 patent, for a simple data item (a data item whose size is less than a particular given size) (S216 and S218), a data identifier (True Name) is computed using a function (e.g., a message digest ("MD") function, such as MD4 or MD5, or a secure hash algorithm ("SHA") function). *Id.* at 14:24-50, 15:37-48, figs. 10(a) & 10(b). As a result, a data item that has an arbitrary length is reduced to a relatively small, fixed size identifier (True Name) that represents the data item. *Id.*

If the data item is a compound data item (a data item whose size is greater than the particular given size), the system will partition the data item into segments (S220); assimilate each segment (S222); compute the True Name of the segment; create an indirect block consisting of the computed segment True Names (S224); assimilate the indirect block (S226); and replace the final 32-bits of the resulting True Name by the length modulo 32 of the compound data item (S228). *Id.* at 15:49-67, fig. 10(b). The result is the True Name of the compound data item. *Id.*

Figure 11 of the '539 patent is reproduced below:

FIG. 11

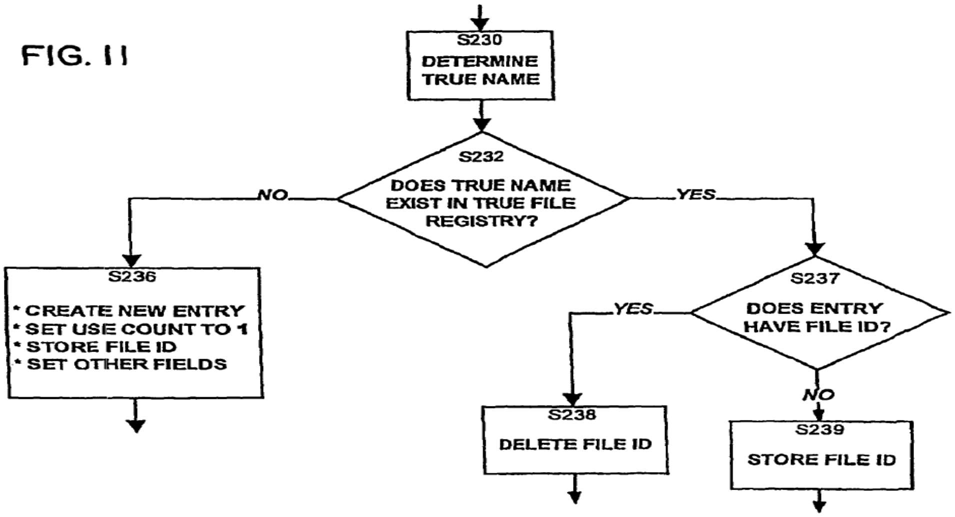


Figure 11 of the ’539 patent depicts a mechanism for assimilating a data item into a file system. The purpose of this mechanism is to add a given data item to the True File registry. *Id*. at 16:10-16. If the data item already exists in the registry, the duplicate will be eliminated. *Id*.

To assimilate a data item, the system will determine the True Name of the data item corresponding to the file (S230); look for an entry for the True Name in the True File Registry (S232); and determine whether a True Name entry exists in the True File Registry (S232). *Id*. at 16:10-29, fig. 11. If the entry record includes a corresponding True File ID (Step S237), the system will delete the file (Step S238). *Id*. Otherwise, the system will store the True File ID in the entry record (S239). *Id*. If there is no entry in the True File Registry for the True Name (S232), the system will create a new entry in the True File Registry for the True Name (S236). *Id*.

C. Illustrative Claim

All of the challenged claims are independent claims. Claim 21 is illustrative and reproduced as follows:

> 21. A computer-implemented method of obtaining access to a data item at a first computer in a network of computers, said data item comprising a plurality of segments, each of said plurality of segments being stored on at least one of a plurality of computers in said network, said plurality of computers being distinct from said first computer, the method comprising the steps of:
>
> (A) by hardware in combination with software, using a first data identifier to obtain a plurality of segment identifiers, each of said segment identifiers corresponding to one of said plurality of segments, the segment identifier for each particular

> segment being based at least in part on a first given function of the data comprising said particular segment and only the data in said particular segment, where any two identical segments will have identical segment identifier as determined using said first given function, and
>
> wherein said first data identifier is based, at least in part, on a second given function of data comprising the plurality of segment identifiers;
>
> (B) using the plurality of segment identifiers obtained in step (A) to obtain at least one of said plurality of segments by, for at least one particular segment identifier of said plurality of segment identifiers:
>
> (b0) using said particular segment identifier to ascertain one or more locations in said network of computers that should have the corresponding particular segment;
>
> (bl) using said particular segment identifier to request said corresponding particular segment from at least one of said one or more locations ascertained in step (b0); and
>
> (b2) obtaining said corresponding particular segment from at least one location in said network.

Ex. 1001, 42:52-43:18.

D. Prior Art Relied Upon

EMC relies upon the following prior art references:

Woodhill	US 5,649,196[2]	July 15, 1997	(Ex. 1005)
Fischer	US 5,475,826[3]	Dec. 12, 1995	(Ex. 1036)

[2] Woodhill claims the benefit of U.S. patent application No. 08/085,596, filed on July 1, 1993.

[3] Fischer was filed on Nov. 19, 1993.

Albert Langer, "*Re: dl/describe (File descriptions)*," posted to the "alt.sources" "comp.archives.admin" newsgroups on Aug. 7, 1991 ("Langer," Ex. 1003)

Frederick W. Kantor, "*FWKCS (TM) Contents_Signature System Version 1.22*," FWKCS122.REF (Aug. 10, 1993) ("Kantor," Ex. 1004)

E. Grounds of Unpatentability

The Board instituted the instant trial based on the following grounds of unpatentability:

Claim	Basis	References
10 and 21	§ 102(b)	Langer
34	§ 103(a)	Langer and Woodhill
10 and 21	§ 103(a)	Kantor
34	§ 103(a)	Kantor and Langer
10 and 21	§ 103(a)	Woodhill and Fischer

II. ANALYSIS

A. Claim Construction

We begin our analysis by determining the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007).

The parties proposed a claim construction for each of the following claim terms: (1) "data" and "data item," (2) "data identifier," (3) "True Name," and (4) "location." Pet. 5-6; Prelim. Resp. 3-5. In the Decision on Institution, we addressed the parties' proposed claim constructions and set forth the broadest reasonable interpretation of each of the claim terms. Dec. 8-13. Neither party challenges our claim constructions. PO Resp. 1-2; Reply in general. We discern no reason to deviate from those constructions for the purposes of this decision. For convenience, the claim constructions proffered in the Decision on Institution are set forth in the table below.

Claim Terms	**Claim Constructions**
data item	Sequence of bits which includes one of the following: (1) the contents of a file; (2) a portion of a file; (3) a page in memory; (4) an object in an object-oriented program; (5) a digital message; (6) a digital scanned image; (7) a part of a video or audio signal; (8) a directory; (9) a record in a database; (10) a location in memory or on a physical device or the like; and (11) any other entity which can be represented by a sequence of bits. Dec. 8-10.
data	A subset of a data item. *Id*. at 9-10.
data identifier	A substantially unique alphanumeric label for a particular data item. *Id*. at 10-11.
True Name	A substantially unique alphanumeric label for a particular data item. *Id*. at 12-13.
location	Any of a particular processor in the system, a memory of a particular process, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any physical location in the system. *Id*. at 13.

After institution, PersonalWeb asserts that the preambles of claims 10 and 21 are limiting. PO Resp. 2-3. PersonalWeb argues that each claim body refers back to its preamble for completeness. *Id*. We agree. In general, a preamble is construed as a limitation "if it recites essential structure or steps, or if it is 'necessary to give life, meaning, and vitality' to the claim." *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 808 (Fed. Cir. 2002) (quoting *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305 (Fed. Cir. 1999)). When the limitations in the body of the claim "rely upon and derive antecedent basis from the preamble, then the preamble may act as a necessary component of the claimed invention." *Eaton Corp. v. Rockwell Int'l Corp.*, 323 F.3d 1332, 1339 (Fed. Cir. 2003).

Notably, the preamble of claim 10 recites "[a] computer-implemented method of obtaining access to *a data item* at a first computer in *a network of computers*, said data item comprising *a plurality of segments*." Ex. 1001, 41:57-60. The preamble of independent claim 21 recites similar features. Indeed, the body of claim 10, which includes the claim terms "said data item," "said network of computers," and "said plurality of segments," relies upon and derives antecedent basis from the preamble of claim 10. Similarly, the body of claim 21, which includes the claim terms "said network of computers" and "said plurality of segment," relies upon and derives antecedent basis from the preamble of claim 21.

Accordingly, we conclude that the preambles of claims 10 and 21 are entitled to patentable weight.

B. Whether Kantor and Langer are "Printed Publications"

In its petition, EMC takes the position that Kantor and Langer each are a "printed publication" within the meaning of 35 U.S.C. § 102(b). Pet. 35-36, 42. EMC asserts that Kantor has been publicly available since August 1993, which is prior to the critical date, April 11, 1995, one year before the earliest priority date claimed by the '539 patent. *Id.* at 4, n.3. EMC also submits that Langer has been publicly available before the critical date, because Langer was made available on the "alt.sources.d" and "comp.archives.admin" Usenet newsgroups on August 7, 1991. *Id.* at 3, n.2, 36. As support, EMC proffers declarations of Mr. Michael A. Sussell (Ex. 1053), Mr. Jason S. Sadofsky (Exs. 1081, 1091), and Mr. Keith Moore (Ex. 1059) to confirm the publication and authenticity of Kantor and Langer.

PersonalWeb counters that neither Kantor nor Langer is a "printed publication." PO Resp. 54-60. In particular, PersonalWeb alleges that EMC has not established that the references existed prior to the critical date, because EMC's witnesses did not review the references before the critical date. *Id.* at 55-57. PersonalWeb also contends that there is no evidence that the references were disseminated publicly, catalogued, or indexed in a meaningful way. *Id.* PersonalWeb maintains that EMC fails to establish that one with ordinary skill in the art, exercising reasonable diligence, would have located the documents prior to the critical date. *Id.*

Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. Rather, we determine that EMC has demonstrated sufficiently that Kantor and Langer are "printed publications."

The determination of whether a given reference qualifies as a prior art "printed publication" involves a case-by-case inquiry into the facts and circumstances surrounding the reference's disclosure to members of the public. *In re Klopfenstein*, 380 F.3d 1345, 1350 (Fed. Cir. 2004). The key inquiry is whether the reference was made "sufficiently accessible to the public interested in the art" before the critical date. *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989); *In re Wyer*, 655 F.2d 221, 226 (CCPA 1981). "A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it . . . ." *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006) (citation omitted).

Indexing is not "a necessary condition for a reference to be publicly accessible," but is only one among many factors that may bear on public accessibility. *In re Lister*, 583 F.3d 1307, 1312 (Fed. Cir. 2009). In that regard, "while often relevant to public accessibility, evidence of indexing is not an absolute prerequisite to establishing online references . . . as printed publications within the prior art." *Voter Verified, Inc. v. Premier Election Solutions, Inc.*, 698 F.3d 1374, 1380 (Fed. Cir. 2012).

Contrary to PersonalWeb's assertion that Kantor did not exist prior to the critical date and there is no evidence that Kantor was disseminated publicly, Kantor itself shows a copyright date of "1988-1993" and a posted date of "1993 August 10." Ex. 1004, Title Page, the first page after the Title

Page ("All of the programs and documents, comprising the entire contents of this Authenticity Verification Zipfile FWKCS122.ZIP, together with this Zipfile itself, are, in accordance with their respective dates of creation or revision, (C) Copyright Frederick W. Kantor 1988-1993."). Kantor also states:

> The FWKCS(TM) Contents_Signature System has become a robust platform for supporting contents_signature functions. FWKCS provides many functions and options for application in a public, commercial, school, institutional, or governmental environment. Extensive technical support is of special value in helping such users to benefit more fully from these many features.
>
> Registered FWKCS hobby BBS users are able to receive a modest amount of assistance, and are invited to participate in the FWKCS conference on The Invention Factory BBS, echoed via Execnet.
>
> Commercial, school, institutional, and governmental users, with their special support needs, are invited to discuss terms for obtaining such assistance.
>
> . . . .
>
> To get a new version of FWKCS, download FWKCSnnn.ZIP from The Invention Factory BBS, where nnn is the new version number without a decimal point. These special downloads are available at no fee, from a 43_line hunt_up group of USR Dual Standard modems, at 2400-16800 bits/sec (including V32.bis).

Ex. 1004, 158-59. It is clear from Kantor that, during the 1988-1993 timeframe, Dr. Kantor had posted many versions of his software and user manual—including Kantor (version 1.22), the version relied upon by EMC (Ex. 1004)—on electronic Bulletin Board Systems.

Mr. Sussell, the co-owner and system operator of the Invention Factory Bulletin Board System, testifies that the Invention Factory Bulletin Board System is a computer system that allows users to share files, messages, and articles, as well as search, upload, and download files. Ex. 1053 ¶¶ 3, 4. According to Mr. Sussell, he and his wife launched the Invention Factory Bulletin Board System in 1983, and it had over 3,000 subscribers by mid-1993. *Id*. at ¶ 6. Mr. Sussell testifies that, by 1993, the system provided all users keyword search functionality and access to various descriptive and meaningful directories. *Id*. at ¶¶ 8-10.

Importantly, Mr. Sussell testifies that the Invention Factory Bulletin Board System "extensively utilized and hosted current versions of FWKCS software on its [Bulletin Board System]," and "made publicly accessible and available the complete FWKCS ZIP file that contained both the software as well as related documentation such as user manuals" prior to the critical date. *Id*. at ¶ 15; *see also id*. at ¶¶ 16-27. Specifically, Mr. Sussell testifies that users would have found Kantor by performing keyword searches on the Invention Factory Bulletin Board System. *Id*. at ¶ 21. Mr. Sussell also indicates that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about the software on the "Welcome" screen, and made the FWKCS Zip file available in four different directories. *Id*. at ¶¶ 18-20. Mr. Sussell further testifies that computer disks that contain the FWKCS Zip file were distributed at various Bulletin Board System conferences. *Id*. at ¶ 18.

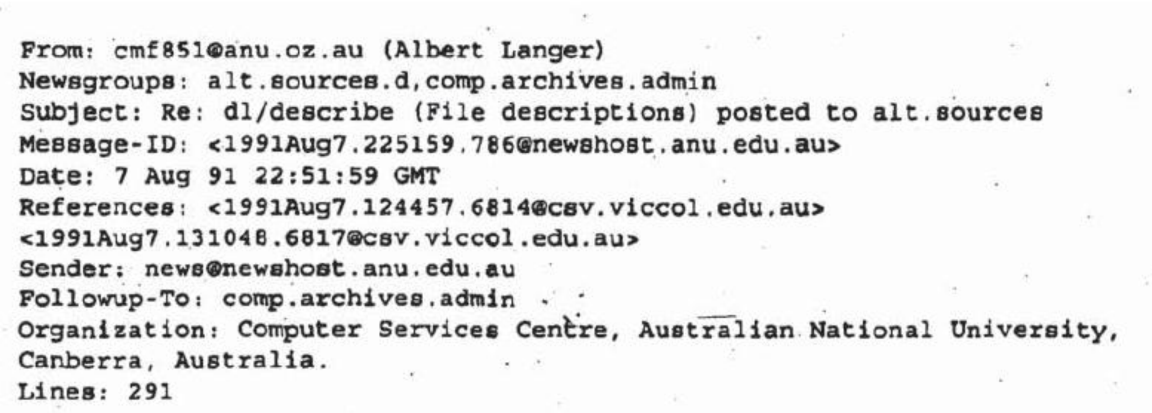
Mr. Sadofsky, a technology archivist and software historian, testifies that he personally verified the authenticity of Kantor—the user manual (version 1.22), the version relied upon by EMC (Ex. 1004)—by comparing it with a "1993 archived" version, and determined that Kantor is identical to the "1993 archived" version. Ex. 1081 ¶¶ 14-17. Mr. Sadofsky testifies that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id*. at ¶ 16; *see also* Ex. 1091 ¶¶ 10-11; Ex. 2014 ¶ 5. According to Mr. Sadofsky, Kantor was publicly accessible prior to the critical date. Ex. 1081 ¶¶ 13, 16-17.

PersonalWeb also asserts that Kantor was buried and hidden in the zip file in a manner such that "it would not have been located and accessed by persons interested and ordinarily skilled in the art exercising reasonable diligence even if they had access to the ZIP file." PO Resp. 58-59 (citing Ex. 2014). However, PersonalWeb's supporting evidence, Mr. Thompson's declaration (Ex. 2013, 2014), does not substantiate PersonalWeb's assertion. Upon review of Mr. Thompson's declaration, we observe that Mr. Thompson downloaded the FWKCS Zip file without any difficultly. Ex. 2014 ¶ 5. Significantly, Mr. Thompson did not follow the instructions provided with the zip file, nor did he use the appropriate computer environment (DOS 3.0 or an IBM OS/2 2.0) that was used normally in 1993-1994 timeframe. Ex. 2014 ¶¶ 6-11; Ex. 1091 ¶¶ 5, 14. Instead, he used non-compatible software (DOS 8.0 and 32-bit Windows XP operating system that was released in 2001). *Id.* Once he followed the instructions and unzipped the FWKCS Zip file, Mr. Thompson located Kantor without difficulty. Ex. 2014 ¶¶ 20-22.

Mr. Sadofsky confirms that the README.TXT file provides simple instructions and, if a user follows the instructions and uses the operating system that was used normally in 1993-1994 timeframe, the user could locate Kantor without difficulty. Ex. 1091 ¶¶ 13-17. In fact, Mr. Sadofsky demonstrated, in his declaration, several relatively easy ways for a user to access Kantor—with or without installing the software, and with or without help screens. Ex. 1091 ¶¶ 8-16 (II. README.TXT); ¶¶ 17-20 (III. GETLOOK.BAT); ¶¶ 21-22 (IV. FWKCS122 Start Screen and In-Program Help). Based on the evidence before us, we determine that Kantor was available to the extent that persons interested and ordinarily skilled in the art, exercising reasonable diligence, could locate it.

The evidence on this record sufficiently supports that Kantor was posted on a publicly accessible site—the Invention Factory Bulletin Board System—well known to those interested in the art and could be downloaded and retrieved from that site, and, therefore, Kantor, an electronic publication, is considered a “printed publication” within the meaning of 35 U.S.C. § 102(b). *See Wyer*, 655 F.2d at 226 (An electronic publication, including an on-line database or Internet publication, is considered to be a “printed publication” “upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it and recognize and comprehend therefrom the essentials of the claimed invention without need of further research or experimentation.”).

Similarly, the evidence on this record shows Langer was publicly distributed prior to the critical date. The header on the first page of Langer, reproduced below, indicates that Langer was distributed on August 7, 1991 to the newsgroups “alt.sources.d” and “comp.archives.admin” (Ex. 1003, 1):



```
From: cmf851@anu.oz.au (Albert Langer)
Newsgroups: alt.sources.d,comp.archives.admin
Subject: Re: dl/describe (File descriptions) posted to alt.sources
Message-ID: <1991Aug7.225159.786@newshost.anu.edu.au>
Date: 7 Aug 91 22:51:59 GMT
References: <1991Aug7.124457.6814@csv.viccol.edu.au>
<1991Aug7.131048.6817@csv.viccol.edu.au>
Sender: news@newshost.anu.edu.au
Followup-To: comp.archives.admin
Organization: Computer Services Centre, Australian National University,
Canberra, Australia.
Lines: 291
```

Mr. Moore, who has personal knowledge of the operation of Usenet in 1991, testifies that Langer’s header is consistent with the format of Usenet articles from the 1991 time frame, and the “Date:” field—indicating that Langer was posted on August 7, 1991, at approximately 10:51 PM GMT—would have generated automatically when the article was posted to Usenet. Ex. 1059 ¶ 16. Mr. Moore also testifies that he personally verified the authenticity of Langer by comparing it with an archived version obtained from Google Groups. *Id.* at ¶ 19.

According to Mr. Moore, Usenet was a network of computers that individuals could use to send and receive technical articles. *Id.* at ¶ 13. In particular, Mr. Moore indicates that anyone could subscribe to a Usenet newsgroup without restrictions, and that subscribers could read articles from the Usenet newsgroups. *Id*. at ¶ 21. Mr. Moore testifies that Langer was distributed through two specific Usenet newsgroups: (1) "alt.sources.d," which hosted technical discussions about source code; and (2) "comp.archives.admin," which focused on technical issues related to the administration of computer archives. *Id*. at ¶ 22. Mr. Moore further declares that Usenet articles were distributed automatically to the registered readers, and, during the 1991-1992 timeframe, the "alt.sources.d" and "comp.archives.admin" newsgroups had 37,000 and 27,000 registered readers, respectively. *Id*. at ¶¶ 23-24. Given the evidence before us, we determine that EMC has established sufficiently that Langer was distributed publicly to those interested in the art.

We also are not persuaded by PersonalWeb's argument that EMC's witnesses personally did not post or review Kantor and Langer prior to the critical date. PO Resp. 55-57 (citing Ex. 2015, 52-55; Ex. 2013, 29-30; Ex. 2016, 98, 180; Ex. 2019, 49-50). It is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986) (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible"); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'"); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would know of the invention.).

For the foregoing reasons, we determine that EMC has demonstrated, by a preponderance of the evidence, that Kantor and Langer are "printed publications" within the meaning of 35 U.S.C. § 102(b). Therefore, EMC may rely upon Kantor and Langer for its asserted grounds of unpatentability under 35 U.S.C. §§ 102(b) and 103(a).

C. Principles of Law

To establish anticipation, each and every element in a claim, arranged as recited in the claim, must be found in a single prior art reference. *Net MoneyIN, Inc. v. VeriSign, Inc*., 545 F.3d 1359, 1369 (Fed. Cir. 2008); *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001). We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994) (citation and internal quotation marks omitted). Moreover, "it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." *In re Preda*, 401 F.2d 825, 826 (CCPA 1968).

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) secondary considerations of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). The level of ordinary skill in the art is reflected by the prior art of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995); *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978).

We analyze the instituted grounds of unpatentability in accordance with the above-stated principles.

D. Claims 10 and 21 – Anticipated by Langer

EMC asserts that claims 10 and 21 are unpatentable under 35 U.S.C. § 102(b) as anticipated by Langer. Pet. 35-40. As support, EMC provides detailed explanations as to how each claim element, arranged as recited in the claim, is disclosed by Langer. *Id*. EMC also relies upon the declaration of Dr. Douglas W. Clark. Ex. 1009 ¶¶ 26-29.

Langer

Langer discloses a method of accessing files in a network of computers. Ex. 1003, 3. For instance, a file request may be embedded in a news article and include a unique identifier for the file. *Id*. at 3-4. As a result, users are informed automatically about the nearest location of the file. *Id*. Langer further discloses that a unique identifier for a file is calculated using a hash function (e.g., MD5, a cryptographic hash function) on the entire contents of the file, rather than the file's location. *Id*. at 2-3. For a package (e.g., an archive, which is a collection of files packaged together) that is divided into its component files, a unique identifier for each component file is calculated by using an MD5 hash function on the contents of the component file. *Id*. at 5. The unique identifier for the entire package is calculated by applying *an MD5 hash again to the concatenation of the MD5 hashes* of the component files ("a hash of hashes"). *Id*.

Discussion

In its patent owner response, PersonalWeb counters that Langer does not describe obtaining a plurality of segment identifiers, or a segment, in response to a request comprising the first identifier, as required by claims 10 and 21. PO Resp. 40-42. As support, PersonalWeb proffers a declaration of Dr. Robert B. K. Dewar. Ex. 2020 ¶¶ 75-78.

Claim 10 recites "in response to a request, said request comprising a first identifier, obtaining a plurality of segment identifiers, . . . using at least one of said segment identifiers . . . , requesting at least one particular

segment of said plurality of segments that comprise said data item."
Ex. 1001, 41:61-42:10. Claim 21 recites similar limitations.

In its petition, EMC takes the position that Langer meets the limitation because Langer describes a method that, in response to a request including the MD5 hash code of the package, obtains the MD5 hash codes for the inner files of the package. Pet. 39 (citing Ex. 1009 ¶ 29; Ex. 1003, 3-5). As support, Dr. Clark testifies that a person with ordinary skill in the art would have understood that, in order to retrieve a particular inner file of the package, the MD5 hash code of the entire package could be used to obtain the MD5 hash codes that were computed for the inner files. Ex. 1009 ¶ 29 (citing Ex. 1003, 4).

PersonalWeb does not disagree that: (1) Langer's package is a data item; (2) the individual inner files of a package are segments; (3) the MD5 codes of the inner files are the segment identifiers; and (4) an MD5 code of the concatenation of the codes of the inner files from the package is the first identifier. PO Resp. 40 (citing Pet. 39; Ex. 1046, 3-6).

However, PersonalWeb alleges that Langer fails to disclose: (1) accessing an inner file of a package by sending a request that includes an MD5 code of the package; (2) obtaining a plurality of MD5 codes of the inner files in response to such a request; and (3) using one of MD5 codes to obtain a particular inner file of the package. *Id*. at 41 (citing Ex. 2020 ¶¶ 75, 76). PersonalWeb contends that EMC improperly attempts to switch back and forth between Langer's package embodiment (Ex. 1003, 5-6) and Langer's standalone file embodiment (Ex. 1003, 3-4). PO Resp. 42. In particular, PersonalWeb maintains that Langer's package embodiment does not meet the disputed limitation because the technique of using MD5 codes is not used to access an inner file of a package, and that Langer's standalone file embodiment also does not meet the disputed limitation because it does not have a plurality of segment identifiers. *Id*. at 42 (Ex. 2020 ¶¶ 76, 77).

In its reply, EMC maintains that Langer explicitly discloses that a user may submit a query to a database using an MD5 code to determine the location of a file so it could be retrieved. Reply 8 (citing Ex. 1003, 3-4). EMC asserts that the MD5 code of a package may be used to obtain the concatenated block of MD5 codes, or a listing of MD5 codes and filenames of the inner files of the package. *Id*. (citing Ex. 1003, 5-6; Ex. 1092 ¶¶ 49, 50; Ex. 1088, 381-82). According to EMC, in either situation, an MD5 code for a particular inner file of the package may then be used to identify and retrieve that particular inner file. *Id*. (citing Ex. 1003, 3-4; Ex. 1009 ¶¶ 28, 29). We agree with EMC.

We observe that PersonalWeb's arguments and expert testimony essentially rest on the incorrect premise that Langer has two separate and distinct embodiments. We disagree with PersonalWeb's assertion that Langer's disclosure under the heading of "Unique Identifiers" (Ex. 1003, 3-4) is a standalone file embodiment, and has little to do with other portions of Langer. In fact, that disclosure of Langer merely teaches the overall concept of utilizing unique identifiers (e.g., MD5 hash codes) to access files. Nothing in Langer indicates that the unique identifiers (Ex. 1003, 3-4) are limited to standalone files, and could not apply to files within a package.

Moreover, we are not persuaded by PersonalWeb's argument that Langer teaches away from the disputed limitation (PO Resp. 41-42). At the outset, we note that, although a "teaching away" argument could be relevant to an obviousness analysis, "whether a reference teaches away from an invention is inapplicable to an anticipation analysis." *ClearValue, Inc. v. Pearl River Polymers, Inc.*, 668 F.3d 1340, 1344 (Fed. Cir. 2012) (quoting *Celeritas Techs., Ltd. v. Rockwell Int'l Corp.*, 150 F.3d 1354, 1361 (Fed. Cir. 1998)) (internal quotation marks omitted).

In any event, PersonalWeb's argument and expert testimony contradict Langer's explicit disclosure. Notably, Dr. Dewar testifies that a person with ordinary skill in the art would have recognized that the MD5 codes of the inner files "would have been *calculated locally* upon receipt of a new package in order to allow a user to see if the codes for a new package matched those of another package – not for accessing files of a package." Ex. 2020 ¶ 76 (emphasis added). As noted by EMC, however, such local calculation would require *downloading the entire package*, contrary to Langer's stated objective for his invention. Reply 9. Indeed, Langer discloses that the files are distributed among different locations, and acknowledges the inefficiency of obtaining the entire package when a user has a new MD5 code for a package. Ex. 1003, 5 ("[I]t would be nice to be able to tell the user *without the need for collecting the entire package*." (emphasis added)). Therefore, in the absence of an explicit disclosure, one with ordinary skill in the art would not have read Langer to download the entire package and calculates the MD5 codes *locally*, as PersonalWeb alleges.

On the other hand, we are persuaded by EMC's contention that an MD5 code for a particular file within a package may be used to identify and retrieve that particular file. As Dr. Clark testifies, Langer discloses that, in response to a request for a file from a user, the central database server uses the MD5 code to return the locations that store a copy of the file corresponding to the MD5 code. Ex. 1009 ¶ 29. Dr. Clark's testimony is consistent with the express disclosure of Langer, which provides that "[a]n archie or similar lookup could first determine which nearby systems have the file," and "that database lookup may as well also provide the local directory and filename for it." Ex. 1003, 4.

Langer also explains that a package's unique identifier is computed by hashing the MD5 codes for the individual files within the package. Ex. 1003, 5. Langer describes a technique that eliminates the need for the user to download the entire package even in the situation where a package has a new MD5 code, and an individual file in a package can be requested. *Id*. at 5-6. Langer further describes that the central database (e.g., archie) explodes the contents of package files and lists the individual items within them. *Id*. at 5. According to Dr. Clark, the MD5 code of the package is used to obtain a listing of MD5 codes and filenames of the inner files in the package. Ex. 1092 ¶ 50 (citing Ex. 1088, 381-82). We credit Dr. Clark's testimony in that regard because it is consistent with Langer's disclosure. Ex. 1003, 5. Therefore, we agree with EMC that an MD5 code for a particular file within the package may then be used to identify and retrieve that particular file. Reply 8 (citing Ex. 1003, 3-4; Ex. 1009 ¶¶ 28, 29).

For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that claims 10 and 21 are anticipated by Langer.

E. Claim 34 – Obvious Over Langer and Woodhill

EMC asserts that claim 34 is unpatentable under 35 U.S.C. § 103(a) as obvious over Langer and Woodhill. Pet. 35-41 (citing Ex. 1003, 3-5; Ex. 1005, 15:13-20). In support of that asserted ground of unpatentability, EMC provides explanations as to how each claim limitation is taught or suggested by the combination of Langer and Woodhill, and a rationale for combining the references. *Id*. EMC also relies upon its explanations regarding the anticipation ground of unpatentability based on Langer, and Dr. Clark's testimony. *Id*.; Ex. 1009 ¶¶ 28-31.

In its patent owner response, PersonalWeb counters that the obviousness ground of unpatentability does not cure the deficiencies of Langer. PO Resp. 43. PersonalWeb essentially relies upon the same arguments presented with respect to the anticipation ground of unpatentability as to claims 10 and 21. *Id*. As discussed above, we have addressed those arguments and determined that they are unavailing. PersonalWeb further argues that the combination of Langer and Woodhill does not disclose all of the limitations of claim 34, and alleges that there is insufficient reason to combine Langer and Woodhill.

Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. In the analysis below, we focus on the deficiencies alleged by PersonalWeb with respect to the ground of unpatentability based on the combination of Langer and Woodhill.

Dividing a data item into a plurality of segments

Claim 34 recites "dividing a particular data item into a plurality of segments; . . . determining a plurality of segment identifiers by . . . determining a corresponding segment identifier for each particular segment of said plurality of segments." Ex. 1001, 45:5-9.

In its petition, EMC maintains that dividing a file into a plurality of segments was a known, effective technique to handle large files, as evidenced by Woodhill, to reduce the amount of data that must be transmitted. Pet. 40-41 (citing Ex. 1009 ¶¶ 30-31; Ex. 1005, 15:13-20). According to EMC, one with ordinary skill in the art would have found it obvious to combine Langer's method of accessing files using unique identifiers (e.g., MD5 codes) and Woodhill's technique of dividing a file into a plurality of segments. *Id*.

In its patent owner response, PersonalWeb counters that Langer fails to disclose the aforementioned limitation recited in claim 34. PO Resp. 43-44. In particular, PersonalWeb argues that Langer does not determine segment identifiers for *all segments* of a package, because "no hash function or algorithm is applied to any directory, directory tree, or header of any package in Langer." *Id*. 44-45 (citing Ex. 2020 ¶ 84). PersonalWeb further alleges that Langer teaches away from the subject matter of claim 34, as

Langer specifically states that it intentionally does not apply a hash function to the directory. *Id*. at 46 (citing Ex. 1003, 5; Ex. 2020 ¶ 85).

We are not persuaded by PersonalWeb's arguments and supporting evidence. As EMC points out (Reply 4-6, 9-10), PersonalWeb's arguments are based on an unreasonable construction of the claim term "data item" that requires the data item to include directories and headers, which are not part of the *contents* of the inner files of a package. As discussed above, the broadest reasonable interpretation of the claim term "data item" is "a sequence of bit." We clarified in this decision and in the Decision on Institution (Dec. 8-10) that one of the examples of a data item is "a *portion* of a file." That example is consistent with the specification of the '539 patent. Ex. 1001, 2:16-21 ("[A] data item may be . . . a portion of a file."). Moreover, PersonalWeb consistently has agreed with that claim interpretation and example. Prelim. Resp. 3; PO Resp. 1-2. Therefore, claim 34 does not require a data item to include directories or headers of a package. Consequently, Langer's technique of determining unique identifiers by applying a hash function to the *contents* of the inner files, and not to directories or headers, describes the aforementioned limitation recited in claim 34.

Furthermore, Langer's technique does not criticize, discredit or otherwise discourage the aforementioned claim limitation. *See DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1327 (Fed. Cir. 2009) ("A reference does not teach away, however, if it merely expresses a general preference for an alternative invention but does not 'criticize, discredit, or otherwise discourage' investigation into the invention claimed.") (quoting *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004)). Rather, we determine that Langer's technique meets the claim limitation, because the claim limitation does not require the data item or segments of the data item to include directories or headers.

Reasons to Combine Langer and Woodhill

In its patent owner response, PersonalWeb argues that it would not have been obvious to one with ordinary skill in the art to combine Woodhill and Langer. PO Resp. 47 (citing Ex. 2020 ¶ 87). In particular, PersonalWeb alleges that Woodhill's system is concerned with backing up files, but Langer never discloses any desire to back up files to a remote backup server. *Id*. PersonalWeb also contends that there is no reason to apply Woodhill's granularization technique that is related to large database files to Langer. *Id*.

We are not persuaded by PersonalWeb's arguments, as they improperly assume that Woodhill's disclosure is limited to backing up files and Langer's method is limited to small non-database files. Indeed, Langer does not place any requirement on the type or size of files. Ex. 1003, 3-5. The mere fact that the two references have different objectives does not mean that a person of ordinary skill in the art would not combine their teachings. *In re Heck*, 699 F.2d 1331, 1333 (Fed. Cir. 1983) ("The use of patents as references is not limited to what the patentees describe as their own inventions or to the problems with which they are concerned.") (quoting *In re Lemelson*, 397 F.2d 1006, 1009 (CCPA 1968)).

Importantly, a prior art reference must be considered for everything it teaches by way of technology and is not limited to the particular invention it is describing and attempting to protect. *EWP Corp. v. Reliance Universal Inc.*, 755 F.2d 898, 907 (Fed. Cir. 1985), *cert. denied*, 474 U.S. 843 (1985). EMC's proposed modification does not require incorporating Woodhill's *entire back-up procedure* into Langer's method. EMC merely relies upon Woodhill's *technique of dividing files* into a plurality of segments. *See KSR*, 550 U.S. at 420 ("[F]amiliar items may have obvious use beyond their primary purpose, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle.").

Upon reviewing the record before us, we determine that EMC's suggestion for modifying Langer's method of accessing a file using an unique identifier with Woodhill's technique of dividing a file into a plurality of segments—to reduce the amount of data that must be transmitted and to provide a more efficient method of handling large data files—suffices as an articulated reason with a rational underpinning to justify the legal conclusion of obviousness. *See KSR*, 550 U.S. 416 ("The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.").

Secondary Considerations of Nonobviousness

PersonalWeb argues that its evidence of non-obviousness outweighs EMC's evidence of obviousness. PO Resp. 54. In support of its argument, PersonalWeb proffers three licensing agreements, as well as the declaration of Mr. Kevin Bermeister. *Id.* (citing Exs. 2010- 12; Ex. 2009 ¶¶ 3-9).

PersonalWeb argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between the challenged claims of the ’539 patent and the above-identified license agreements. Reply 14. EMC argues that each of the licenses granted rights to more than just the challenged claims, and involved related parties with interlocking ownership and business interests. *Id*. We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and the challenged claim.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *GPAC*, 57 F.3d at 1580. Further, without a showing of nexus, “the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness” when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000); *Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus. Neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010-12) are directed to the claimed subject matter recited in any of the challenged claims. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in any of the challenged claims. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Furthermore, even if we assume that above-identified licenses establish some degree of industry respect for the claimed subject matter recited in challenged claims, that success is outweighed by the strong evidence of obviousness over the combination of Langer and Woodhill presented by EMC.

Based on this record, including the evidence of obviousness and the secondary considerations regarding licensing activities, we conclude that EMC has demonstrated by a preponderance of the evidence that claim 34 would have been obvious over the combination of Langer and Woodhill.

F. Claims 10 and 21 – Obvious Over Kantor

EMC asserts that claims 10 and 21 are unpatentable under 35 U.S.C. § 103(a) as obvious over Kantor. Pet. 42-49 (citing Ex. 1009 ¶¶ 31, 34-37, 39, 41-43; Ex. 1004, Preface, 7-11, 48, 51-55, 96-97, 173-74; Ex. 1047). As support, EMC provides explanations as to how each claim limitation is taught or suggested by Kantor. *Id*. EMC also relies upon Dr. Clark's testimony. *Id*.

PersonalWeb counters that Kantor does not disclose segment identifiers, as recited in claims 10 and 21. PO Resp. 3-14. PersonalWeb also argues that there is insufficient reason to modify Kantor's commands to permit identifying files based on contents signatures. *Id*. at 15-22.

Kantor

Kantor describes a method of identifying duplicate files, by using contents signatures that are generated based on the contents of the files, instead of the file names or file locations. Ex. 1004, 2-4, 6-8, 48-49. In particular, Kantor applies a hash function (e.g., a cyclic residue check or cyclic redundancy check ("CRC")) to each file to obtain the contents signature for each file. *Id.* at 6-8, 48-49. For each zip file, Kantor creates zip-file contents signatures by hashing the contents signatures for the files contained within the zip file ("a hash of hashes"). *Id.* at 2, 9. As described by Kantor, this is done by "adding together all the 32_bit CRC's for the files in the zipfile, modulo 2^32, separately adding together their uncompressed file_lengths modulo 2^32, and then arranging the two resulting hexadecimal number as a single structure." *Id.* at 9. Kantor stores the contents signatures and zip-file contents signatures in a master contents-signature list (e.g., CSLIST.SRT). *Id.* at 18.

According to Kantor, contents signatures and zip-file contents signatures are useful for identifying files that have the same contents stored on the electronic bulletin board systems. Ex. 1004, 2 of Preface, 5, 9. For example, when uploading a zip file, the system determines whether that zip file already exists in the system using the zip-file contents signature, and determines whether the inner files of that zip file already exist in the system using the contents signatures for the inner files. *Id.* at 9. Kantor specifically acknowledges the benefits of using contents signatures and zip-file contents signatures to: (1) find files or zip files on the system and delete duplicate files or zip files uploaded under different names; and (2) determine whether a collection of files that corresponds to one zip file is contained in a larger zip file or spread among several different zip files. *Id*.

Segment Identifiers

Claim 10 recites "said data item comprising a plurality of segments . . . the segment identifier for each particular segment being based, at least in part, on a first given function of the data comprising said particular segment and only the data in said particular segment." Ex. 1001, 41:59-67. Claim 21 also recites a similar limitation. In its petition, EMC asserts that this limitation is met by Kantor's technique of calculating the contents signatures for the inner files of a zip file by applying a function to the inner files. Pet. 43-45 (citing Ex. 1004, Preface, 7-9; Ex. 1009 ¶¶ 34-35).

PersonalWeb responds that Kantor teaches away from the claimed subject matter, because Kantor applies the CRC hash function to *uncompressed* files before they are compressed and packaged into the zip file. PO Resp. 4-13 (citing, *e.g.*, Ex. 2016, 65, 67; Ex. 2020 ¶¶ 25-26). According to PersonalWeb, the CRC hash function is applied to different bit sequences (uncompressed files) than the bit sequences (compressed files) that make up the inner files of the zip file. *Id*.

We are not persuaded by PersonalWeb's arguments, as they are based incorrectly on the assumption that Kantor's inner files of a zip file must be *compressed files*. Rather, we agree with EMC that zip files can have *uncompressed* inner files. Reply 1 (citing Ex. 1092 ¶¶ 6-9; Ex. 1088, 263-64). Indeed, PersonalWeb does not disagree that zip files *are not* *always compressed*. Ex. 1088, 263-64. As PersonalWeb's evidence shows, the standard zip-file format, used by Kantor at the time of the invention, defines seven compression methods, which include "Compression method 0" that *does not compress the inner files* when packaging them into a zip file. Ex. 2004, 3; Ex. 1004, 2 of Preface.

Dr. Dewar's reliance on Kantor's statements regarding file compression ratio to support his testimony—"Kantor confirms that the 'files' in the ZIP files described in Kantor are compressed"—is misplaced. Ex. 2020 ¶ 28 (citing Ex. 1004, 2 of Preface, 9, 55). The mere fact that Kantor refers to a compression ratio does not support PersonalWeb's position that the inner files of a zip file must be compressed, because in the situation where "Compression method 0" is used—which *does not compress the inner files*—the file compression ratio is one. Contrary to Dr. Dewar's testimony, those portions of Kantor cited by Dr. Dewar do not require each inner file of a zip file to be compressed. Instead, the cited portions of Kantor merely state that the zip-file contents signature *depends on the contents of the inner files*, and provide examples of items that the zip-file contents signature do not depend upon. *See, e.g.*, Ex. 1004, 2 of Preface ("FWKCS has the special ability to make a 'zipfile contents signature', ('zcs') which is *independent of* . . . the names and dates of files in the zipfile, zipped path information, and file compression ratio."); *id*. at 9 ("This has the desirable property that the resulting zcs *does not depend* on the names of the files, . . . nor on the method nor amount of compression . . . ." (emphasis added)).

As EMC notes, even if Kantor only used compressed inner files, Kantor still would describe the disputed claim limitation—"a first given function of the *data* comprising said particular segment and only the *data* in said particular segment"—as the first given function would include a function that *hashes and compresses the data* in the file. Reply 1 (citing Ex. 1092 ¶ 12). Indeed, nothing in the claim language limits a given function to just *hashing* the data. PersonalWeb does not explain adequately why a given function cannot comprise both *hashing and compressing the data*. Moreover, *compressing* a file merely changes *the format* of the file, but it does not change *the contents* of the file. In other words, both compressed and uncompressed versions of an inner file have the *same contents* (i.e., the data). As discussed above, Kantor's contents signatures are generated based on *the contents* of the files (Ex. 1004, 6-8), and Kantor's zip-file contents signatures depend on *the contents* of the inner files and do not depend on the *format* of the inner files (Ex. 1004, 2 of Preface, 9, 55). Claims 10 and 21 do not place any limitation on *the format* of the plurality of segments.

PersonalWeb's "teaching away" argument is misplaced. Even if Kantor expresses a general preference for applying a function to uncompressed files to obtain contents signatures before they are compressed, that itself does not operate to criticize, discredit, or otherwise discourage investigation into the aforementioned claim limitation. *See DePuy Spine*, 567 F.3d at 1327. As discussed above, Kantor's contents signatures meet the claim limitation regardless of whether the inner files are compressed or non-compressed.

PersonalWeb also argues that, because a zip file includes additional information (e.g., headers) and Kantor does not obtain a content signature for *headers*, Kantor fails to describe a segment identifier for each segment of the data item. PO Resp. 5-6, 13. That argument is unpersuasive, as it is not commensurate with the scope of claims. *See In re Self*, 671 F.2d 1344, 1348 (CCPA 1982) (It is well established that limitations not appearing in the claims cannot be relied upon for patentability.). Indeed, because claims 10 and 21 each recite the open-ended phrase "comprising" when describing what a data item includes (a "data item *comprising* a plurality of segments"), a data item is not limited to just a plurality of segments, and may include additional information (e.g., headers). Furthermore, as discussed above, the claim term a "data item" includes "a *portion* of a file." Consequently, the inner files meet the claim term "a plurality of segments," and Kantor's technique of generating contents signatures by applying a function to the inner files describes the disputed claim limitation.

Contrary to PersonalWeb's argument that Kantor merely "reads" the CRC values (PO Resp. 9-10), Kantor determines a zip-file contents signature by "adding together all the 32_bit CRC's for the files in the zip file, modulo 2^32, separately adding together their uncompressed_file_lengths modulo 2^32, and then arranging the two resulting hexadecimal numbers as a single structure." Ex. 1004, 9. Dr. Clark testifies that addition module 2^32 is a well-known simple hashing function that uses addition to calculate a value for a file based on the contents of the file. Ex. 1009 ¶ 35 (citing Ex. 1011). Dr. Clark's testimony is consistent with Kantor's disclosure that the resulting zip-file contents signature depends on the contents of the inner files and "does not depend on the names of the files, the dates of the files, the order in which they appear in the zip file, nor on the method nor amount of compression, nor does it depend on comments." Ex. 1004, 3, 9.

Given Kantor's express disclosure and the evidence before us, we determine that EMC has demonstrated sufficiently that Kantor describes segment identifiers for a plurality of segments as recited in claims 10 and 21.

Identifying files using contents signatures

Claim 10 recites "using at least one of said segment identifiers . . . requesting at least one particular segment of said plurality of segments . . . obtaining said particular segment from said at least one of a plurality of computers in said network of computers." Ex. 1001, 42:6-12.

In its petition, EMC recognizes that the users typically request files based on the file names. Pet. 45. Nonetheless, EMC asserts that a person having ordinary skill in the art would have found it obvious to modify the electronic Bulletin Board Systems commands, including the download and read commands, to identify files using contents signatures or zip-file contents signatures, instead of file names. *Id*. at 46 (citing Ex. 1009 ¶ 41). According to EMC, "this would facilitate integrity checking by more precisely specifying the file of interest by its content, and thus improve accuracy." *Id*. Dr. Clark testifies that such a modification would provide a more efficient and context-free means for accessing and sharing files. Ex. 1009 ¶ 41.

PersonalWeb counters that it would not have been obvious to modify Kantor so that the read and download requests would accept contents signatures to identify files. PO Resp. 15-22. In particular, PersonalWeb argues that Kantor teaches away from replacing conventional file names with contents signatures for identifying files, because "Kantor intentionally designed his contents-signatures so that certain different files would have the same signature." PO Resp. 16-18 (citing Ex. 1004, 3, 51; Ex. 2020 ¶¶ 44-46). PersonalWeb also alleges that Kantor fails to teach or suggest the alleged modification, and fails to provide any suggestion or motivation for the alleged modification. *Id.* at 18-22 (citing Ex. 2020 ¶¶ 47-48). PersonalWeb submits that Kantor does not disclose any problems with the use of conventional file names for the read and download requests. *Id.* at 22.

We are not persuaded by PersonalWeb's arguments. First, PersonalWeb's teaching away argument is misplaced, as it fails to recognize that the cited portion of Kantor specifically explains that the different files that allegedly have the same signature files also have the *same contents*. *See* Ex. 1004, 3 ("[T]he same file contents . . . will have the same zipfile contents signature."). In fact, that is one of the reasons why using contents signatures or zip-file contents signature, instead of file names, to identify files is more accurate. Ex. 1004, Preface, 5, 9. Notably, files that have the *same contents* would be identified as duplicates, and files that have *different contents* would be identified as different files, regardless of whether they have different file names. *Id.* As Kantor notes, finding and deleting duplicate files would improve system efficiency. *Id.*

Further, PersonalWeb's argument that Kantor does not teach or suggest the alleged modification is unpersuasive, because an obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418. PersonalWeb's argument overlooks "the fundamental proposition that obvious variants of prior art references are themselves part of the public domain." *Translogic,* 504 F.3d at 1259. Moreover, we observe that the asserted ground of unpatentability is based on the *combination* of Kantor's teaching of using contents signatures to identify files with Kantor's teaching of requesting files. It is well settled that nonobviousness cannot be established by attacking each prior art teaching individually where, as here, the ground of unpatentability is based upon a combination of different teachings in the prior art. *See In re Keller,* 642 F.2d 413, 426 (CCPA 1981). Rather, the test for obviousness is whether the combination of prior art teachings, taken as a whole, would have suggested the patentees' invention to a person having ordinary skill in the art. *See In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

As to PersonalWeb's arguments that Kantor does not provide a motivation for the modification (PO Resp. 18; Ex. 2020 ¶ 47), a rationale to combine the prior art teachings does not have to be found explicitly in the prior art, itself. *See In re Kahn,* 441 F.3d 977, 987 (Fed. Cir. 2006) (A "motivation to combine the relevant prior art teachings does not have to be found explicitly in the prior art."). We also are not persuaded by PersonalWeb's argument that there would have not been a logical reason to modify Kantor in the manner alleged by EMC, other than impermissible hindsight (PO Resp. 22). As discussed above, EMC asserts that it would have been obvious to modify the read and download commands to identify files using contents signatures instead of file names. Pet. 46 (citing Ex. 1009 ¶ 41). EMC takes the position that "this would facilitate integrity checking by more precisely specifying the file of interest by its content, and thus improve accuracy." *Id.* Dr. Clark testifies that such a modification would provide a more efficient and context-free means for accessing and sharing files. Ex. 1009 ¶ 41. EMC's position and Dr. Clark's testimony are consistent with Kantor's disclosure that using contents signatures, instead of file names, to find and delete duplicate files would increase system efficiency by reducing storage cost and system time for locating and managing files. Ex. 1004, Preface, 5, 9, 205-206. As such, we conclude that EMC has articulated a sufficient reason to combine the teachings of Kantor.

We are not persuaded by PersonalWeb's arguments that the proposed modification is not enabled and that EMC fails to explain how the proposed modification could have been carried out to yield a predictable result. PO Resp. 19-22. EMC specifically explains that Kantor's Precheck and Lookup operations provide examples of user commands that utilize contents signatures. Pet. 46 (citing Ex. 1004, 97, 173; Ex. 1009 ¶¶ 37, 39, 41). For instance, Kantor describes the Precheck operation as a software utility running on the electronic Bulletin Board Systems for identifying files that already uploaded in the system by using their contents signatures. Pet. 46-47 (citing Ex. 1004, 173). Dr. Clark explains that Kantor's Lookup operation permits users to submit a request containing a contents signature to determine where the corresponding file is located on the system. Ex. 1009 ¶ 41 (citing Ex. 1004, 96-97). As Dr. Clark also notes, Kantor's "i" function provides users with the capability to submit contents-signature search requests to find files on the system that contain material related to a user's file, by obtaining the contents signatures for the inner files of the zip files that contain the related material. *Id*. (citing Ex. 1004, 96-97). Dr. Clark further testifies the system as modified would have utilized one of those contents signatures for the inner files in a download request to obtain the particular inner file that is associated with the contents signature. *Id*. Upon review of the parties' contentions and supporting evidence, we credit Dr. Clark's testimony as it is consistent with Kantor's disclosure. We also agree with EMC that Dr. Clark merely relies on the disclosure of Kantor (Ex. 1004, 96-97), and not LOOKUP.DOC and PRECHECK.DOC files as alleged by PersonalWeb. For the foregoing reasons, we determine that EMC has explained sufficiently how the proposed modification could have been carried out to yield a predictable result.

In addition, PersonalWeb agrees that, at the time of the invention, users on the electronic Bulletin Board Systems had the capability to request a file using the *file name*. PO Resp. 18. In light of Kantor, a person of ordinary skill in the art would have recognized how to calculate contents signatures and zip-file contents signatures and how to use them to identify files. *See, e.g.*, Ex. 1004, Preface, 5-9. A person with ordinary skill in the art also would have appreciated the benefit of using contents signature and zip-file contents signatures that are generated based on the contents of the files, rather than *file names*, for identifying files accurately. *Id*. The mere substitution of contents signatures and zip-file contents signatures for *file names* in read and download requests predictably uses prior art elements according to their established functions. Such a substitution is an obvious improvement. *See KSR*, 550 U.S. at 417 (The simple substitution of one known element for another is likely to be obvious if it does no more than yield predictable results.). Moreover, PersonalWeb has not provided sufficient evidence that such a substitution is beyond the level of a person with ordinary skill in the art. *See Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007).

Conclusion

We also are not persuaded by PersonalWeb's evidence of non-obviousness, because it fails to establish the required nexus, as discussed above. For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that claims 10 and 21 are unpatentable over Kantor.

G. Claim 34 – Obvious Over Kantor and Langer

EMC asserts that claim 34 is unpatentable under 35 U.S.C. § 103(a) as obvious over Kantor and Langer. Pet. 42-50 (citing Ex. 1003, 3-5; Ex. 1004, Preface, 7-11, 48, 51-55, 96-97, 173-74). EMC maintains that Kantor, as understood by a person with ordinary skill in the art, renders claim 34 obvious. *Id*. at 42-49. However, EMC relies upon Langer to teach a cryptographic hash function in case the claim term "a True Name of the data" is limited to use of such a function. *Id*. at 49-50. EMC also relies upon its explanations regarding the obviousness ground of unpatentability based on Kantor, itself, as to claims 10 and 21, and Dr. Clark's testimony. *Id*. at 42-49 (citing Ex. 1004, Preface, 7-11, 48, 51-55, 96-97, 173-74).

PersonalWeb counters that Langer does not cure the deficiencies of Kantor. PO Resp. 22-39. PersonalWeb further argues that the combination of Kantor and Langer does not disclose all of the limitations recited in claim 34. *Id*. PersonalWeb also relies upon some of the same arguments presented with respect to claims 10 and 21. *See, e.g.*, *id.* at 35-37.

Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. In the analysis below, we focus on the deficiencies alleged by PersonalWeb as to the ground of unpatentability based on the combination of Kantor and Langer, and we address each of PersonalWeb's argument in turn.

Dividing a data item into a plurality of segments

Claim 34 recites "dividing a particular data item into a plurality of segments; . . . determining a plurality of segment identifiers by . . . determining a corresponding segment identifier for each particular segment of said plurality of segments." Ex. 1001, 45:5-9. PersonalWeb argues that Kantor fails to disclose that limitation. PO Resp. 23-44.

First, PersonalWeb argues that Kantor does not determine segment identifiers for *all segments* of a zip file. *Id*. at 23-34 (citing Ex. 2020 ¶¶ 50-52). We are not persuaded. As discussed above, an example of a data item is a *portion* of a file. As Dr. Clark testifies (Ex. 1092 ¶¶ 28-33), "a sequence of bits" does not require that the bits must be *contiguous*. We also observe that a large database file—a data item—may be stored across different storage systems or memories. *See, e.g.*, Ex. 1005, 14:53-15:8; Ex. 1003, 5. Therefore, the "dividing" limitation does not require necessarily the zip file*, in its entirety*, to be divided into a plurality of segments. In other words, a plurality segments may include merely the inner files of a zip file, excluding headers and other information about the data.

Next, PersonalWeb alleges that Kantor intentionally designed his contents signatures so that different files would have the same signature. PO Resp. 24. As discussed above, that argument is unpersuasive because it fails to recognize that the cited portion of Kantor specifically explains that the different files that allegedly have the same signature also have the *same contents*. *See* Ex. 1004, 3 ("[T]he same file contents . . . will have the same zipfile contents signature.").

According to PersonalWeb, the disputed claim limitation expressly requires that the step of *determining* the segment identifiers take place *after* the "*dividing*" step. PO Resp. 35 (citing Ex. 2020 ¶¶ 26-32, 62-63). Based on that interpretation, PersonalWeb alleges that Kantor does not meet that limitation, because Kantor determines the *CRC values* before the files are packaged into a zip file. *Id*.

PersonalWeb's argument is not persuasive, as it does not consider the situation in which the new file to be uploaded to the system is already a zip file (Ex. 1004, 9). In that situation, the inner files of such a zip file are packaged in a zip file *before* the CRC values are calculated.

Even applying PersonalWeb's interpretation of the disputed claim limitation and its reading of Kantor, we determine that EMC has established sufficiently that Kantor's *contents signatures* render the disputed claim limitation obvious. We observe that a *CRC value* is not the same as the *contents signature* (segment identifier) of an inner file. That is an important distinction. The *contents signature* (segment identifier) for each inner file is generated by using the *CRC value and the length value* of the file. Ex. 1004, 8. Kantor discloses that the system looks inside (i.e., *unzips*) the zip file, and then uses the information (e.g., the *CRC values*) in the zip file to generate the *contents signatures* (segment identifiers) for the inner files in the zip file. Ex. 1004, 48. According to Kantor, this is a relatively quick operation, as the zip file already contains the *CRC values*. *Id*. Therefore, even if, as alleged by PersonalWeb, *the CRC values* are calculated before the files are packaged into a zip file, Kantor explicitly discloses that the *contents signatures* (segment identifiers) for the inner files are calculated *after* the inner files are packaged into a zip file. Ex. 1004, 8, 48-49.

For the foregoing reasons, we determine that EMC has demonstrated sufficiently that the combination of Kantor and Langer teaches or suggests the "dividing" claim limitation.

Segment identifier being a True Name

Claim 34 recites "the segment identifier for each particular segment being a True Name of the data comprising said particular segment." Ex. 1001, 45:10-12. As discussed above, the claim term "True Name" is construed as "a substantially unique alphanumeric label for a particular data item." PersonalWeb alleges that the combination of Kantor and Langer does not describe that limitation. PO Resp. 35-37. PersonalWeb relies upon the same arguments presented with respect to claims 10 and 21. *Id*. As discussed above, those arguments are unpersuasive.

Determining a True Name of the second data item

Claim 34 recites "forming a second data item comprising said plurality of segment identifiers, . . . in response to a request to access said data item, said request comprising said data item identifier, providing at least said second data item." Ex. 1001, 46:1-10.

In its patent owner response, PersonalWeb alleges that the combination of Kantor and Langer does not describe the aforementioned limitation. PO Resp. 37-39. Specifically, PersonalWeb assumes that the second data item is made up of Kantor's *CRC values,* and based on that assumption, it argues the following: (1) Kantor fails to disclose arranging the *CRC values* in a sequence of bits (*id*. at 38); (2) the zip-file contents signature is not a True Name of a sequence of the *CRC values* (*id*. at 39); and (3) Kantor fails to disclose providing all the *CRC values* in response to a request to access the data item that includes the data item identifier (*id*.). We are not persuaded by PersonalWeb's arguments.

We instead agree with EMC that PersonalWeb's assumption that the second data item is made up of Kantor's *CRC values* is incorrect. Reply 7. PersonalWeb's argument, once again, ignores the important distinction between the *CRC values* and the *contents signatures* for the inner files. As EMC notes, Kantor explicitly discloses that the *contents signatures* of the inner files are stored in a master contents-signature list. Pet. 45 (citing Ex. 1004, 18). Further, Dr. Clark explains that Kantor's Lookup *command* (Ex. 1004, 173) which uses the "remote Inquiries" option (the "i" function) provides users with the capability to submit contents-signature search *requests* to find files on the system that contain material related to a user's file. Ex. 1009 ¶ 41 (citing Ex. 1004, 96-97); *see also* Pet. 47. Dr. Clark also testifies that, in response to a Lookup request including a zip-file contents signature when using the "y form of the TEST" function, the system provides the user *the full set of contents signatures* (a second data item) for all the inner files in each of the zip files in which the specific file appears. Ex. 1092 ¶ 41 (citing Ex. 1004, 96-98). We credit Dr. Clark's testimony as it is consistent with the explicit disclosure of Kantor.

Conclusion

We also are not persuaded by PersonalWeb's evidence of non-obviousness, because it fails to establish the required nexus, as discussed above. For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that claim 34 is unpatentable over the combination of Kantor and Langer.

H. Claims 10 and 21 – Obvious Over Woodhill and Fischer

EMC asserts that claims 10 and 21 are unpatentable under 35 U.S.C. § 103(a) as obvious over Woodhill and Fischer. Pet. 50-57 (citing Ex. 1009 ¶¶ 44-59). EMC acknowledges that Woodhill's disclosure of restoring a file does not use a hash of the granule identifiers to identify a database file. Pet. 50-55. Nevertheless, EMC indicates that using a "*hash of hashes*" technique for identifying database or compound files was known in the art at the time of the invention, as evidenced by Fischer (Ex. 1036, 7:49-8:38). Pet. 56.

PersonalWeb counters that the combination of Woodhill and Fischer does not describe certain claim limitations, and it would not have been obvious to combine Woodhill and Fischer in the manner asserted by EMC. PO Resp. 47-54. Upon review of the evidence on record, we are not persuaded by PersonalWeb's arguments.

Woodhill

Woodhill discloses a system for distributed storage management using binary object identifiers. Ex. 1005, 1:11-17. The system includes a remote backup file server in communication with a plurality of local area networks. *Id*. Woodhill's system includes a Distributed Storage Manager ("DSM") program for building and maintaining a file database. *Id*. at 3:44-49, fig. 3. The DSM program views a file as a collection of data streams, and divides each data stream into one or more binary objects. *Id*. at 4:13-23, 7:40-43, fig. 5A, item 132. For each binary object being backed up, a binary object identification record is created in a file database and includes a binary object identifier to identify each binary object uniquely. *Id*. at 7:60-8:1, 8:33-34. Binary object identifiers are calculated based on the contents of the data instead of from an external and arbitrary source, such that the binary object identifier changes when the contents of the binary object changes. *Id*. at 8:57-62, 8:40-42. The DSM program also utilizes a technique of subdividing the large database files into granules and then tracks changes from the previous copy of the granules. *Id*. at 14:53-65. This technique is used to reduce the amount of data that must be transmitted to the remote backup file server. *Id*. at 15:4-8. Figure 5I of Woodhill, reproduced below, illustrates the process of restoring a file to a previous version:

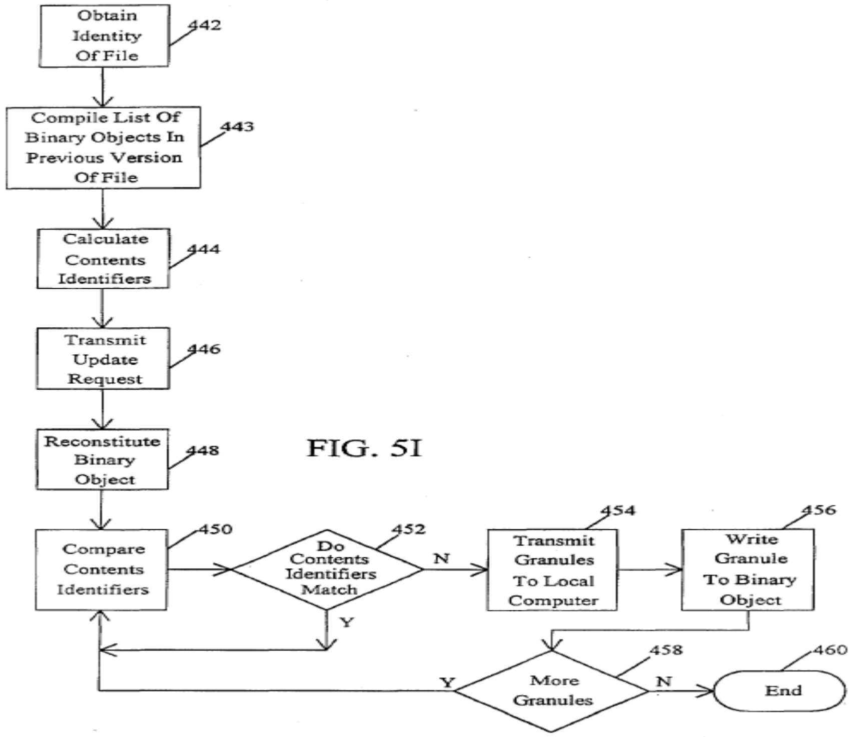

Obtain Identity Of File
442
Compile List Of Binary Objects In Previous Version Of File
443
Calculate Contents Identifiers
444
Transmit Update Request
446
Reconstitute Binary Object
448
Compare Contents Identifiers
450
Do Contents Identifiers Match
452
N
Y
Transmit Granules To Local Computer
454
Write Granule To Binary Object
456
More Granules
458
Y
N
End
460

FIG. 5I

As shown in Figure 5I of Woodhill, in response to a user's request to restore a file, the DSM program restores the previous version of the binary object by retrieving the granules from the remote server. *Id*. at 17:18-18:9. The DSM program uses the previous version of granule contents identifiers to determine the location of the granules. *Id*. at 17:50-55, fig. 5I, box 450. It compares the previous version of contents identifiers with the contents identifiers for the granules within the current version of the file. *Id*. The DSM program locates the granule on the local computer when the contents identifiers match. *Id*. at 17:58-60. If the contents identifiers do not match, the DSM program locates the granule on the remote server and transmits the granule from the remote server to the local computer. *Id*. at 17:60-64, fig. 5I, box 454. After the granules that are located on the remote server have been transferred to the local computer, the file on the local computer is restored to its previous version. *Id*. at 18:6-9.

Using a segment identifier to request a segment

Claim 10 recites "using at least one of said segment identifiers . . . requesting at least one particular segment of said plurality of segments that comprise said data item." Ex. 1001, 42:6-9. Claim 21 recites a similar limitation.

In its petition, EMC relies upon Woodhill's disclosure of restoring a file to meet this limitation. Pet. 54-55 (citing Ex. 1005, 17:18-18:9). Specifically, EMC asserts that Woodhill discloses an *update request* to restore a current version of a binary object to a prior version of a binary object, which includes the binary object identifier for the prior version of the binary object. *Id*. at 54 (citing Ex. 1005, 7:60-8:4, 17:17-50; Ex. 1009 ¶¶ 48, 50-52). According to EMC, in response to the *update request*, the DSM program uses the contents identifiers (segment identifiers) for the granules to obtain a granule (segment) from the remote backup file server for the local computer. *Id*. at 55 (citing Ex. 1005, 17:50-18:9; Ex. 1009 ¶¶ 46, 49, 53).

PersonalWeb counters that Woodhill does not disclose the "request" limitation. PO Resp. 47-51 (citing Ex. 2020 ¶¶ 88-93). In particular, PersonalWeb asserts that Woodhill does not use contents identifiers to request granules of a binary object, because "a contents identifier is never provided in any 'request' for a particular granule." *Id*. at 48-50 (citing Ex. 2020 ¶¶ 90-93). According to PersonalWeb, the contents identifiers are compared to determine whether to transmit granules, but are not used for *requesting* granules. *Id*. at 51 (citing Ex. 1005, 17:51-65; Ex. 2020 ¶ 92).

We are not persuaded by PersonalWeb's arguments, as they are not commensurate with the scope of claims 10 and 21. Rather, we agree with EMC that the claim limitation does not require a content identifier (segment identifier) to be *provided in a request* for a particular granule (segment). *See* Reply 11-12 (citing Ex. 1005, 17:18-46; Ex. 1009 ¶¶ 48-54). Claims 10 and 21 merely require *using* a segment identifier to request a particular segment. In fact, PersonalWeb's expert acknowledges that "the contents identifiers are *used* to determine which granules have changed via the comparison," and to identify which granule is to be transmitted from the remote backup file server to the local computer. PO Resp. 49-50; Ex. 2020 ¶ 92 (emphasis added). Further, Woodhill discloses that an *update request* includes the binary object identifiers for the binary object of the previous version of the file, as well as granule contents identifiers (segment identifiers) for each granule of the current version (segment). Ex. 1005, 17:36-46; Ex. 1088, 185. As discussed above, during the comparison of the granule contents identifiers, if the contents identifiers do not match, the DSM program locates the particular granule on the remote server and transmits the granule from the remote server to the local computer. *Id*. at 17:60-64, fig. 5I, box 454. In other words, the DSM program requests the particular granule using its contents identifiers from the remote server.

We also are not persuaded by PersonalWeb's argument that "there is no disclosure in Woodhill of using a Binary Object Identifier 74 to obtain a plurality of contents identifiers." PO Resp. 51 (citing Ex. 2020 ¶ 93). Instead, we agree with EMC that the binary object identifier is used to obtain its corresponding granule content identifiers. Reply 12 (citing Ex. 1005, fig. 3; Ex. 1009 ¶ 52; Ex. 1088, 185, 196-197). Woodhill discloses that, in response to the user's request to restore a file to the previous version, the DSM program compiles a list of all binary objects comprising the current version of the user-specified file from the file database, which includes the binary object identifiers of all the binary objects for the file. Ex. 1005, 17:27-36, fig. 3. The DSM program then calculates the *contents identifiers* for granules within the current version of each binary object. *Id*. at 17:36-40. Given the evidence before us, we determine that EMC has demonstrated sufficiently that the combination of Woodhill and Fischer would have rendered the "request" claim limitation obvious.

First identifier is based on a second function of segment identifiers

Claim 10 recites "said first identifier is based, at least in part, on a second given function of the plurality of segment identifiers." Ex. 1001, 42:2-5. Claim 21 recites a similar limitation. EMC acknowledges that Woodhill's disclosure of restoring a file does not use a hash of the granule identifiers to identify the database file that contains the granules. Pet. 50-55. Nevertheless, EMC indicates that using a "*hash of hashes*" technique for identifying database or compound files was well known, as evidenced by Fischer (Ex. 1036, 7:49-8:38). *Id*. at 56. EMC contends that a person of ordinary skill in the art would have modified Woodhill's file restoring process by calculating the identifier for the large database file based on a function of the granule contents identifiers ("a hash of hashes"), as taught by Fischer, because this would improve the efficiency and performance of Woodhill's data processing for restoring a file. *Id*. at 57 (citing Ex. 1009 ¶ 59). As support, Dr. Clark testifies that "if only a few granules are changed, it is faster to compute a hash of the granule identifiers (rather than of the entire binary object) because the previously calculated granule identifiers could be re-used." Ex. 1009 ¶ 59. In response, PersonalWeb advances three arguments to support its contention that it would not have been obvious to modify Woodhill. PO Resp. 51-54.

1. *Hashing contents identifiers*

PersonalWeb argues that the binary object identifiers could not have been generated based on contents identifiers, because the contents identifiers do not exist when the binary object identifiers are calculated. *Id*. at 52 (citing Ex. 2020 ¶ 96). However, EMC counters that PersonalWeb's assumption that the binary object identifiers must be calculated *before* contents identifiers are determined is incorrect, because Woodhill does not impose such a requirement. Reply 12-13 (citing Ex. 1092 ¶¶ 68-69).

We agree with EMC that Woodhill does not require any specific order for calculating binary object identifiers and contents identifiers. The portion of Woodhill cited by PersonalWeb does not support its assumption. PO Resp. 52 (citing Ex. 2020 ¶ 96; Ex. 1005, 17:44). As Dr. Clark testifies, the calculation of contents identifiers does not depend on a binary object identifier. Ex. 1092 ¶ 68 (Ex. 1005, 5:12-9:28, 14:52-18:9). Dr. Clark also testifies that the cited portion of Woodhill merely demonstrates that the binary object identification records for the *previous version* of the binary object exist at the time an update request for restoring the previous version is made, and the contents identifiers calculated in step 444 of Woodhill's Figure 5I are for the granules within the *current version* of each binary object as it exists on the local computer. Ex. 1092 ¶ 69. We credit Dr. Clark's testimony as it is consistent with Woodhill's disclosure. *See* Ex. 1005, 5:12-9:28, 14:52-18:9, 17:18-50.

2. *Teaching away argument*

PersonalWeb argues that Woodhill teaches away from the modification. PO Resp. 52-53 (citing Ex. 2020 ¶ 97). According to PersonalWeb, Woodhill teaches that Fischer's technique is undesirable, because Woodhill emphasizes that the binary object identifier is calculated from the *contents* of the data instead of from an external and arbitrary source, whereas Fischer calculates the "fileHash" *using external and arbitrary sources*. *Id*. (citing Ex. 1005, 8:40-42; Ex. 1036, 8:4-55). EMC counters that PersonalWeb's argument is based on the incorrect assumption that the record identifiers "K" of Fischer's fileHash must be related to *external* information. Reply 13. Dr. Clark testifies that PersonalWeb's reliance on the statement of Fischer (Ex. 1036, 5:55-58) is incorrect, because an "employee number" corresponds to exactly one employee record and, therefore, is *neither external nor arbitrary*. Ex. 1092 ¶ 71.

We agree with EMC that Fischer does not require the record identifiers to be based on *external* information. In fact, in the same sentence relied upon by PersonalWeb, Fischer discloses that "a record number [] K_i *may* be any indexing value." Ex. 1036, 5:55-58 (emphasis added). Even if Fischer's technique requires a calculation using an *external* source, obviousness does not require that all of the features of the secondary reference be bodily incorporated into the primary reference. *In re Etter*, 756 F.2d 852, 859 (Fed. Cir. 1985) (en banc); *Keller*, 642 F.2d at 425.

In any event, EMC's proposed modification does not change Woodhill's process for calculating the contents identifiers which are based on the *contents* of the granules. Pet. 55-57. By applying "a hash of hashes" technique, the binary object identifiers for the database file would be calculated using a hash function (second function) against the *contents identifiers* of the granules associated with the binary objects. *Id*. Such binary object identifiers still would be based on the *contents* of the binary object, as Woodhill's "key notion" statement suggests. Ex. 1005, 8:38-42.

We are not persuaded by PersonalWeb's argument that Woodhill is concerned with uniquely identifying a binary object, whereas Fischer is concerned with security. The mere fact that the two references have different objectives does not mean that a person with ordinary skill in the art would not combine their teachings. *Heck*, 699 F.2d at 1333 ("The use of patents as references is not limited to what the patentees describe as their own inventions or to the problems with which they are concerned." (citation and internal quotation marks omitted)); *EWP Corp.*, 755 F.2d at 907.

3. *Argument regarding inoperability*

PersonalWeb asserts that the alleged combination would have rendered Woodhill's system inoperable for its intended purpose. PO Resp. 53-54 (Ex. 2020 ¶ 98). Specifically, PersonalWeb argues that many parts of Woodhill's system rely on binary object identifiers to detect changes in binary objects that are not granularized, and, thus, one of ordinary skill in the art would not have modified Woodhill's binary object identifiers to base them on granule contents identifiers because this would have resulted in Woodhill's system being inoperative. *Id*.

However, EMC's proposed modification is limited to Woodhill's restoring process in which a binary object has been divided into a plurality of *granules,* and is not limited to the non-granularization situations, as alleged by PersonalWeb. Pet. 54-56 (citing Ex. 1005, 17:7-50, 17:60-18:4). The binary object identifiers for the database file being restored would be calculated based on a function of *the content identifiers of the granules.* Pet. 56. Moreover, as Dr. Clark explains, Woodhill recognizes that a binary object identifier may be calculated in various ways. Ex. 1009 ¶ 72 (citing Ex. 1005, 8:38-40). Dr. Clark further testifies that, regardless of whether a binary object identifier is calculated directly from the contents of the binary object, *or* it is calculated as a function of granule contents identifiers for the granules associated with the binary object, the binary object identifier is based on *the contents of the binary object*. *Id.* Therefore, we are not persuaded that EMC's proposed modification would render Woodhill's system inoperable for its intended purpose. *Keller*, 642 F.2d at 425 ("The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference . . . .").

Conclusion

We also are not persuaded by PersonalWeb's evidence of non-obviousness, because it fails to establish the required nexus, as discussed above. For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that claims 10 and 21 are unpatentable over Woodhill and Fischer.

I. EMC's Motion to Exclude

EMC seeks to exclude: (1) three license agreements (Exs. 2010-12); (2) Mr. Kevin Bermeister's declarations (Exs. 2009, 2022) relating to those license agreements; and (3) Mr. Todd Thompson's declaration (Ex. 2014). Paper 59 ("Pet. Mot."). PersonalWeb filed the license agreements and

Mr. Bermeister's declarations as evidence of non-obviousness to rebut EMC's assertion that claims 10, 21, and 34 would have been obvious over the various combinations of Langer, Kantor, Woodhill, and Fischer. PO Resp. 54. As to Mr. Thompson's declaration, PersonalWeb proffered that evidence to support its assertion that Kantor was not made sufficiently accessible to an interested person. *Id*. at 58-60. PersonalWeb opposes EMC's motion to exclude. Paper 63. In response, EMC filed a reply to PersonalWeb's opposition to its motion to exclude. Paper 66.

With respect to the license agreements and Mr. Bermeister's declarations, EMC argues that they are irrelevant under Federal Rule of Evidence 402[4] and highly prejudicial, confusing, and misleading under Federal Rule of Evidence 403. Pet. Mot. 1-13. As to Mr. Thompson's declaration, EMC argues that it should be excluded under Federal Rule of Evidence 402. *Id.* at 14-15. Specifically, EMC alleges the following: (1) Mr. Thompson does not possess the skill of a person of ordinary skill in the art (*id.* at 14 (citing Ex. 1086, 13-14)); (2) Mr. Thompson did not use compatible software from the relevant time period (*id.* (citing Ex. 1086, 40-41; Ex. 2014, 4, 6)); and (3) Mr. Thompson did not follow the instructions provided with the zip file (*id.* (citing Ex. 1086, 32-35)).

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding PersonalWeb's supporting evidence, we have determined that Kantor is a

[4] As stated in 37 C.F.R. § 42.62, the Federal Rules of Evidence generally apply to *inter partes* reviews.

"printed publication" under 35 U.S.C. § 102(b) and EMC has demonstrated by a preponderance of the evidence that claims 10, 21, and 34 are unpatentable over the various combinations of Langer, Kantor, Woodhill, and Fischer.

Accordingly, EMC's motion to exclude evidence is *dismissed* as moot.

J. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following items of evidence: (1) Kantor (Ex. 1004) and Langer (Ex. 1003); (2) certain documents that corroborate the knowledge and recollections of EMC's witnesses (Exs. 1050-1052, 1055-1058, 1061-1064, 1073, 1077, 1078, 1083-1085), and the portions of testimony regarding these documents; (3) the declarations of Messrs. Sussell, Sadofsky, and Moore (Exs. 1053, 1081, 1091, 1059), and Mr. Sadofsky's deposition (Ex. 2013, 30, 66); and (4) Clark's rebuttal declaration (Ex. 1092). Paper 55 ("PO Mot.").

EMC opposes PersonalWeb's motion to exclude. Paper 64 ("Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 67 ("PO Reply"). For the reasons stated below, PersonalWeb's motion to exclude is *denied*.

1. <u>Kantor and Langer</u>

PersonalWeb alleges that Kantor and Langer should be excluded as unauthenticated and inadmissible hearsay under Federal Rules of Evidence 901 and 902. PO Mot. 1-6. In particular, PersonalWeb argues that "[n]o witness of record has personal knowledge of Kantor or Langer existing prior to [the critical date], and electronic data such as Kantor and Langer is inherently untrustworthy because it can be manipulated from virtually any location at any time." *Id*. at 1. According to PersonalWeb, the dates provided by Kantor and Langer are inadmissible hearsay because Kantor and Langer are not self-authenticating. *Id*.

EMC argues that Kantor and Langer have been authenticated under Federal Rules of Evidence 901, and that the documents are not hearsay, because they are being offered for what they describe—not for the truth of their disclosures. Opp. 1-10. In particular, EMC disagrees with PersonalWeb that the documents cannot be authenticated without direct testimony from a witness with personal knowledge that the documents existed prior to the critical date. *Id*. at 1. EMC asserts that it need "only produce evidence 'sufficient to support a finding' that the reference 'is what the proponent claims it is.'" *Id*. (citing Fed. R. Evid. 901(a)). EMC also contends that testimony from Messrs. Sussell, Sadofsky, and Moore provides sufficient evidence to authenticate Kantor and Langer. *Id*. at 1-10 (citing Exs. 1053, 1081, 1091, 1059).

In its reply, PersonalWeb argues that the Federal Rules of Evidence identified by EMC are not applicable to Kantor and Langer, because Messrs. Sussell, Sadofsky, and Moore did not post or review the documents prior to the critical date. PO Reply 1-5. PersonalWeb also alleges that the authenticity of Kantor and Langer is suspicious, as electronic data is inherently untrustworthy and there is no chain of custody. *Id*.

We are not persuaded that Kantor and Langer should be excluded. At the outset, we disagree with PersonalWeb's position that a witness cannot authenticate a document, unless the witness is the author of the document or the witness has reviewed the document prior to the critical date. Federal Rule of Evidence 901(a) states that the authentication requirement is satisfied if the proponent presents "evidence sufficient to support a finding that the item is what it proponent claims it is." Therefore, neither a declaration from the author, nor evidence of someone actually viewing the document *prior to critical date*, is required to support a finding that the document is what it claims to be. *See Hall*, 781 F.2d at 899 (concluding that "competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible"); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'").

Further, it is well settled that an uninterrupted chain of custody is not a prerequisite to admissibility, but rather gaps in the chain go to weight of the evidence. *U.S. v. Wheeler*, 800 F.2d 100, 106 (7th Cir. 1986); *see also U.S. v. Aviles*, 623 F.2d 1192, 1198 (7th Cir. 1980) ("If the trial judge is satisfied that in reasonable probability the evidence has not been altered in any material respect, he may permit its introduction." (citation omitted)). There is a strong public policy for making all information filed in a quasi-judicial administrative proceeding available to the public, especially in an *inter partes* review, which determines the patentability of a claim or claims in an issued patent. It is within the Board's discretion to assign the appropriate weight to be accorded to evidence.

Although Messrs. Sussell, Sadofsky, and Moore personally did not review Kantor or Langer *prior to the critical date*, they nevertheless have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of the documents. *See Hall*, 781 F.2d at 899; *Wyer*, 655 F.2d at 226; *Bayer*, 568 F.2d at 1361. Notably, Mr. Sussell, the co-founder and system operator of the Invention Factory Bulletin Board System, testifies that Dr. Kantor released his software on the Invention Factory Bulletin Board System in the 1980s, and his system continuously utilized and hosted current versions of the software and user manuals. Ex. 1053 ¶¶ 3, 13, 15. Mr. Sussell also testifies that his system advertised Dr. Kantor's software, and made the software and user manual publicly accessible. *Id*. at ¶ 18. According to Mr. Sussell, his system had over 3,000 subscribers, in the 1993 timeframe, and the users had keyword searching capability to retrieve Kantor. *Id*. at ¶¶ 6, 21.

Although we are cognizant that electronic documents generally are not self-authenticating, it has been recognized that "[t]o authenticate printouts from a website, the party proffering the evidence must produce some statement or affidavit from someone with knowledge of the website . . . for example a web master or someone else with personal knowledge would be sufficient." *St. Luke's Cataract and Laser Institute v. Sanderson*, 2006 WL 1320242, *2 (M.D. Fla. 2006) (quoting *In re Homestore.com, Inc. Sec.Litig.*, 347 F. Supp. 2d 769, 782 (C.D.Cal. 2004)) (internal quotation marks omitted); *see also Market-Alerts Pty. Ltd. v. Bloomberg Finance L.P.*, 922 F. Supp. 2d 486, 493, n.12 (D. Del. 2013) (citing *Keystone Retaining Wall Sys., Inc. v. Basalite Concrete Prods., LLC*, 2011 WL 6436210, *9 n.9 (D. Minn. 2011)) (Documents generated by a website called the Wayback Machine have been accepted generally as evidence of prior art in the patent context); *U.S. v. Bansal*, 663 F.3d 634, 667-68 (3d. Cir. 2011) (concluding that the screenshot images from the Internet Archive were authenticated sufficiently under Federal Rule of Evidence 901(b)(1) by a witness with personal knowledge of its contents, verifying that the screenshot the party seeks to admit are true and accurate copies of Internet Archive's records).

Here, Mr. Sadofsky, who is a technology archivist and software historian, and currently is an archivist for the Internet Archive, testifies that he launched website textfiles.com and subdomain cd.textfiles.com to collect software, data files, and related materials from Bulletin Board Systems. Ex. 1081 ¶¶ 9-11. According to Mr. Sadofsky, textfiles.com and cd.textfiles.com are dedicated to preserving, archiving, and providing free access to unaltered historical software programs and information that initially were made available on the Bulletin Board Systems. *Id.* Mr. Sadofsky states that he previously archived the FWKCS Zip file (FWKCS122.ZIP) that contains Dr. Kantor's software and user manual to cd.textfiles.com from his own copy of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM. *Id.* at ¶ 14 (citing

Ex. 1052). Mr. Sadofsky also testifies that he personally verified the authenticity of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—by comparing it with the "1993 archived" version and determined that Kantor is identical to the "1993 archived" version. Ex. 1081 ¶¶ 13-15. Mr. Sadofsky confirms that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id*. at ¶ 16; Ex. 1091 ¶¶ 10-11; Ex. 2014 ¶ 5. Mr. Sadofsky concludes that Kantor was publicly accessible prior to the critical date. Ex. 1081 ¶¶ 13, 16. PersonalWeb does not present sufficient or credible evidence to the contrary. Based on the evidence before us, we determine that Kantor has been authenticated sufficiently to warrant its admissibility under Federal Rules of Evidence 901(b)(1), (b)(3), and (b)(4).

With respect to Langer, Mr. Moore, who has personal knowledge of the operation of Usenet in 1991, testifies that Langer's header is consistent with the format of Usenet articles from the 1991 time frame, and the "Date:" field—indicating that Langer was posted on August 7, 1991, at approximately 10:51 PM GMT—would have generated automatically when the article was posted to Usenet. Ex. 1059 ¶ 16. Mr. Moore also testifies that he personally verified the authenticity of Langer by comparing it with an archived version obtained from Google Groups, which contains a compilation of Usenet articles going back to the 1980s and is recognized as a key archive of Usenet articles. *Id*. at ¶ 19. Accordingly, Langer has been authenticated sufficiently to warrant its admissibility under Federal Rules of Evidence 901(b)(1), (b)(3) and (b)(4).

We are not persuaded by PersonalWeb's argument that the download date of "7/29/2003" in the lower, right-hand corner calls into question whether Langer existed prior to the critical date. The mere fact that a "downloaded" copy of Langer has a date subsequent to earliest effective filing date is not sufficient to rebut EMC's supporting evidence that Langer is what it claims to be—an article posted on Usenet newsgroups on August 7, 1991. *See, e.g.*, Ex. 1059 ¶¶ 11-17.

Moreover, we agree with EMC that Kantor and Langer also have been authenticated as an "ancient document" under Federal Rule of Evidence 901(b)(8).[5] Opp. 6, 9. Each document is at least 20 years old and can be found in a place where an authentic 20-year old document distributed through a Bulletin Board System or Usenet would likely be. Ex. 1081 ¶¶ 7-8; Ex. 1059 ¶ 19; *see also* Fed. R. Evid. 901(b)(8) 2012 Adv. Comm. Note ("The familiar ancient document rule of the common law is extended to include data stored electronically or by other similar means."). Furthermore, testimony of Messrs. Sussell, Sadofsky, and Moore has established sufficiently that the documents are in a condition that creates no suspicion about their authenticity. Exs. 1053, 1081, 1059. Accordingly, we

[5] Fed. R. Evid. 901(b)(8). Evidence About Ancient Documents or Data Compilations. For a document or data compilation, evidence that it:

(A) is in a condition that creates no suspicion about its authenticity;
(B) was in a place where, if authentic, it would likely be; and
(C) is at least 20 years old when offered.

conclude that Kantor and Langer also have been authenticated sufficiently to warrant its admissibility under Federal Rule of Evidence 901(b)(8).

In addition, we are not persuaded by PersonalWeb's hearsay arguments. As EMC notes, a prior art document submitted as a "printed publication" under 35 U.S.C. § 102(b) is offered simply as evidence of what it described, not for proving the truth of the matters addressed in the document. *See Joy Techs., Inc. v. Manbeck*, 751 F. Supp. 225, 233 n.2 (D.D.C. 1990), *judgment aff'd*, 959 F.2d 226 (Fed. Cir. 1992); Fed. R. Evid. 801(c) 1997 Adv. Comm. Note ("If the significance of an offered statement lies solely in the fact that it was made, no issue is raised as to the truth of anything asserted, and the statement is not hearsay."). Therefore, neither Kantor nor Langer is hearsay under Federal Rule of Evidence 801(c).

We further agree with EMC that the posted and copyright dates set forth in Kantor and Langer are not a basis for excluding the documents, as testimony from Messrs. Sussell, Sadofsky, and Moore sufficiently establishes that Kantor and Langer existed prior to the critical date. Further, the computer-generated timestamp—August 10, 1993, at 1:22 AM—of the "1993 archived" version of Kantor (Ex. 1081 ¶¶ 14-15; Ex. 1091 ¶¶ 10-11; Ex. 2014 ¶ 5) also independently corroborates Kantor's existence as of August 10, 1993. *See, e.g., U.S. v. Khorozian*, 333 F.3d 498, 506 (Fed. Cir. 2003) (concluding that an automatically generated time stamp on a fax was not a hearsay statement because it was not uttered by a person). Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Kantor and Langer as impermissible hearsay.

For the foregoing reasons, we decline to exclude Kantor and Langer.

2. Documents Corroborating Witnesses' Knowledge and Recollections

PersonalWeb asserts that a number of documents submitted by EMC (Exs. 1050-1052, 1055-1058, 1061-1064, 1073, 1077, 1078, 1083-1085), and the declarations of Messrs. Sussell and Sadofsky (Exs. 1053, 1081, 1091) regarding those documents should be excluded, because the documents have not been authenticated properly and are inadmissible hearsay. PO Mot. 6-9. PersonalWeb argues that EMC "has not established that any of these documents existed prior to the critical date, and no witness has personal knowledge of their alleged existence prior to April 11, 1995." *Id*. at 7. PersonalWeb further maintains that the documents that are Exhibits 1056, 1057, 1077, and 1078 are irrelevant, prejudicial, and confusing, as they discuss a version of Kantor different than the version relied upon by EMC (version 1.22, Ex. 1004). *Id*. at 8-9.

EMC responds that its witnesses provided those documents to corroborate their independent knowledge and recollections. Opp. 10. EMC asserts that the documents have been authenticated under Federal Rules of Evidence 901-902, and fall within a hearsay exception under Federal Rules of Evidence 803-807. *Id*. at 10-12. We are persuaded by EMC's arguments.

As the movant, PersonalWeb has the burden of proof to establish that it is entitled to the requested relief. 37 C.F.R. § 42.20(c). As discussed previously, we disagree with PersonalWeb's argument that documents cannot be authenticated without direct testimony from the author or a witness who actually reviewed the documents prior to the critical date.

See Fed. R. Evid. 901(a). Significantly, PersonalWeb's motion does not explain sufficiently why each document should be excluded. For instance, PersonalWeb does not explain adequately why the declaration of Mr. Sussell (Ex. 1053 ¶¶ 3-4, 7, 10-11) is not sufficient to authenticate Exhibits 1055-1058, why the declarations of Mr. Sadofsky (Ex.1081 ¶¶ 3, 4, 6-8; Ex. 1091 ¶¶ 4-9) are not sufficient to authenticate Exhibits 1050-1052 and 1083-1085, or why the declaration of Mr. Moore (Ex 1059, ¶¶ 4-14) is not sufficient to authenticate Exhibits 1061-1064, 1073. *See* Fed. R. Evid. 901(b)(1).[6] Nor does PersonalWeb explain sufficiently why certain documents are not self-authenticated: (1) Exhibits 1055-1057, 1077-1078 – documents that have trade inscriptions; and (2) Exhibit 1052 – a photograph of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM, that has Simtel trade inscriptions. *See* Fed. R. Evid. 902(6)-(7).[7]

In its motion, PersonalWeb also fails to identify, specifically, the textual portions of the aforementioned exhibits that allegedly are being

[6] Fed. R. Evid. 901(b)(1). Testimony of a Witness with Knowledge. Testimony that an item is what it is claimed to be.

[7] Fed. R. Evid. 902. Evidence that Is Self-Authenticating
The following items of evidence are self-authenticating; they require no extrinsic evidence of authenticity in order to be admitted:

. . . .

(6) Newspapers and Periodicals. Printed material purporting to be a newspaper or periodical.

(7) Trade Inscriptions and the Like. An inscription, sign, tag, or label purporting to have been affixed in the course of business and indicating origin, ownership, or control.

offered for the truth of the matter asserted, yet seeks to exclude the entirety of each exhibit. The burden should not be placed on the Board to sort through the entirety of each exhibit and determine which portion of the exhibit PersonalWeb believes to be hearsay. Rather, PersonalWeb should have identified, in its motion, the specific portions of the evidence and provided sufficient explanations as to why they constitute hearsay. Additionally, PersonalWeb does not explain adequately why the declarations of Messrs. Sussell, Sadofsky, and Moore do not provide the proper foundation and corroboration for the documents.

To the extent PersonalWeb relies upon the same arguments with respect to Kantor for excluding the documents, we have addressed those arguments above and determined that they are unavailing. We also agree with EMC that the documents concerning prior versions of Kantor are relevant, and not prejudicial or confusing as alleged by PersonalWeb, because such circumstantial evidence provides context and corroboration for the witnesses' independent knowledge and recollection.

Furthermore, we are not persuaded that the declarations of Messrs. Sussell, Sadofsky, and Moore (Exs. 1053, 1081, 1091, 1059) should be excluded. As we discussed above, and we elaborate below in the next section, Messrs. Sussell, Sadofsky, and Moore have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor and Langer. The documents they cite serve to corroborate their independent knowledge and recollection.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude Exhibits 1050-1052, 1055-1058, 1061-1064, 1073, 1077, 1078, 1083-1085, as well as the declarations of Messrs. Sussell, Sadofsky, and Moore (Exs. 1053, 1081, 1091, 1059), which include testimony concerning those exhibits.

3. Declarations of Messrs. Sussell, Sadofsky, and Moore

PersonalWeb argues that the declarations of Messrs. Sussell, Sadofsky, and Moore (Exs. 1053, 1081, 1091, 1059) should be excluded as hearsay under Federal Rule of Evidence 801 and inadmissible under Federal Rules of Evidence 802-807 for lack of foundation and personal knowledge, and Federal Rule of Evidence 702 as improper testimony, because the witnesses personally did not review Kantor (Ex. 1004), Simtel (Ex. 1052), and Langer (Ex. 1003) prior to the critical date. PO Mot. 9. PersonalWeb also argues that Messrs. Sussell, Sadofsky, and Moore "are not qualified experts" in the field. *Id*. at 10. PersonalWeb further alleges that Mr. Sadofsky's deposition (Ex. 2013, 30, 66) should be excluded, as it was responsive to a leading question (*id*. at 65-66) and non-responsive to another question (*id*. at 30-31). PO Mot. 10-11.

EMC responds that the testimony of Messrs. Sussell and Sadofsky should not be excluded, because their testimony is based on their own personal knowledge and recollection, and the documents they cite serve to corroborate their independent knowledge and recollection. Opp. 12-13. EMC further explains that the witnesses have described thoroughly the underlying facts, and, therefore, the testimony should be admitted as relevant under Federal Rules of Evidence 401-402, supported by personal knowledge and foundation under Federal Rule of Evidence 602, and proper opinion testimony under Federal Rules of Evidence 701-703. We find that EMC's contentions have merit.

PersonalWeb's arguments rest on the erroneous premise that EMC's witnesses must have reviewed Kantor, Simtel, or Langer, personally *prior to the critical date* in order to provide competent testimony regarding Kantor, Simtel, or Langer. As discussed previously, it is well settled that it is not necessary for the witnesses to have reviewed the reference personally *prior to the critical date*, in order to establish publication. *See, e.g., Wyer*, 655 F.2d at 226. Although Messrs. Sussell, Sadofsky, and Moore are not experts related to the claimed subject matter of the '539 patent, each witness nevertheless has sufficient personal knowledge and working experience to provide competent testimony. *See Hall*, 781 F.2d at 899. Mr. Sussell was the co-owner and system operator of the Invention Factory Bulletin Board System from 1983 to 1996. Ex. 1081 ¶ 3. Mr. Sussell's testimony is based on his personal knowledge of the relevant facts related to the Invention Factory Bulletin Board System and its association with Kantor. *Id.* ¶ 2. Notably, Dr. Kantor specifically thanked Mr. Sussell in his user manual for hosting Dr. Kantor's software FWKCS and for Mr. Sussell's role in its development. Ex. 1004, 3 ("To Michael Sussell, sysop of The Invention Factory (R), home board for the support of FWKCS, for bringing the problem of duplicate files to my attention and for his help in testing . . . .").

Mr. Sadofsky is a technology archivist and software historian, and works for the Internet Archive, which provides the Wayback Machine service. Ex. 1081 ¶ 3. Mr. Sadofsky also directed an eight-episode documentary film regarding the Bulletin Board Systems. *Id*. at ¶ 4. Mr. Sadofsky's testimony is based on his personal knowledge of the relevant facts related to Kantor and the "1993 archived" version of Kantor. *Id*. at ¶ 2; Ex. 1091 ¶ 2. For example, Mr. Sadofsky personally verified the authenticity of Kantor by comparing it with the "1993 archived" version, and determined that Kantor is identical to the "1993 archived" version. Ex. 1081 ¶¶ 14, 15.

Similarly, Mr. Moore has personal knowledge and working experience with Usenet in 1991. Ex. 1059 ¶¶ 13-16. Mr. Moore's testimony is based on his personal knowledge of the relevant facts related to Usenet and its association with Langer. *Id*. at ¶ 10. Notably, Mr. Moore was intimately familiar with the operation of Usenet in the 1991-1992 timeframe, and he personally verified the authenticity of Langer by comparing it with an archived version obtained from Google Groups. *Id*. at ¶ 19.

Upon review of the evidence on the record, we also agree with EMC that Messrs. Sussell, Sadofsky, and Moore have disclosed sufficient underlying facts to support their testimony. For instance, the computer-generated timestamp—August 10, 1993, 1:22 AM—associated the "1993 archived" version of Kantor corroborates the testimony of Messrs. Sussell and Sadofsky regarding Kantor's existence as of August 10, 1993. Ex. 1081 ¶¶ 14-15; Ex.1091 ¶¶ 10-11; Ex. 2014 ¶ 5.

As to Mr. Sadofsky's deposition, PersonalWeb does not explain sufficiently why that testimony should be excluded. PO Mot. 11. Moreover, Mr. Sadofsky's deposition (Ex. 2013, 30, 66) is consistent with his direct testimony (Ex. 1081 ¶¶ 14-16), and, therefore, it would not prejudice PersonalWeb even if such evidence is not excluded.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude any portion of the declarations of Messrs. Sussell, Sadofsky, and Moore (Exs. 1053, 1081, 1091, 1059) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66).

4. Clark's Rebuttal Declaration

PersonalWeb asserts that Dr. Clark's rebuttal declaration (Ex. 1092) should be excluded, because it is irrelevant, prejudicial, and confusing, as well as beyond the scope of this proceeding. PO Mot. 11-15. In support of its assertion, PersonalWeb advances several arguments. *Id*.

First, PersonalWeb argues that Dr. Clark's rebuttal declaration cites to references that do not serve as the basis of a ground of unpatentability instituted in this proceeding. *Id*. at 11-12. EMC counters that Dr. Clark's statements referencing those references were offered in response to PersonalWeb's argument that one with ordinary skill in the art would not have modified Kantor or Woodhill. Opp. 13 (citing PO Resp. 15-22, 51-54; Ex. 2020 ¶¶ 42-48, 94-98). According to EMC, those statements are relevant to the instituted grounds of unpatentability and confirm that the use of hash-based identifiers to identify files was well known in the art at the time of invention. *Id*. We agree with EMC that Dr. Clark's statements are proper rebuttal evidence submitted in response to PersonalWeb's arguments. Those references were cited merely to show the knowledge level of a person with ordinary skill in the art. *See Randall Mfg. v. Rea*, 733 F.3d 1355, 1362 (Fed. Cir. 2013) (When considering whether a claimed invention would have been obvious, "the knowledge of [an ordinarily skilled] artisan is part of the store of public knowledge that must be consulted."). Such evidence does not change the combination that formed the basis of the grounds of unpatentability based on obviousness instituted in this proceeding. *Id.*; *see also In re Donohue*, 766 F.2d 531, 534 (Fed. Cir. 1985). Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Dr. Clark's rebuttal declaration.

Second, PersonalWeb contends that the "capable," "can," and "may" statements in Dr. Clark's rebuttal declaration should be excluded, because those statements are irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding. PO Mot. 12. PersonalWeb further submits that Dr. Clark's rebuttal declaration includes new obviousness allegations not presented previously with the petition. *Id*. at 12-13. In response, EMC contends that the statements in Dr. Clark's rebuttal declaration were offered in response to PersonalWeb's arguments. Opp. 13-15 (citing *e.g.*, PO Resp. 9-10; Ex. 2020 ¶ 31-32). Having reviewed PersonalWeb's patent owner response and Dr. Clark's rebuttal declaration, we determine that Dr. Clark's testimony is reasonable rebuttal evidence in light of PersonalWeb's arguments. Furthermore, PersonalWeb's arguments concerning Dr. Clark's statements affect the weight to be given by us to

Dr. Clark's testimony in deciding whether the instituted grounds of unpatentability render the challenged claimed unpatentable. When weighing evidence, we are capable of determining whether the prior art references anticipate or render obvious the challenged claims without being confused, misled, or prejudiced by Dr. Clark's testimony. Thus, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude any portions of Dr. Clark's rebuttal declaration.

Finally, PersonalWeb contends that Dr. Clark's rebuttal declaration contradicts his prior deposition. PO Mot. 13-15. We are not persuaded by PersonalWeb's arguments. Rather, we agree with EMC that Dr. Clark's rebuttal testimony is consistent with his earlier testimony. Opp. 15. For instance, Dr. Clark's rebuttal testimony that "zipfiles are not *always* compressed," and the inner files of a zip file may be *uncompressed* (Ex. 1092 ¶¶ 9-11), is consistent with his earlier testimony that the inner files of a zip file are compressed *typically* (Ex. 2016, 55, 59, 66-67). Moreover, Dr. Clark's testimony is reasonable rebuttal evidence in light of the evidence submitted by PersonalWeb. Dr. Clark merely points out in his rebuttal declaration that PersonalWeb's evidence also shows that zip files are not *always* compressed. Ex. 1092 ¶ 9 (citing Ex. 2004, 3 (the zip file format defines seven compression methods which include "Compression method 0" that does not compress the file); Ex. 1088, 262 (Dr. Dewar agrees that "the zipfile standard allows for uncompressed files.")).

In addition, we agree with EMC that Dr. Clark's testimony does not conflict with EMC's position advanced in its petition that the inner files in

Kantor constitute the relevant portion of the zip file for determining segment identifiers. Opp. 15 (citing Pet. 44; Ex. 1009 ¶ 35). We do not discern that Dr. Clark's answer to a question related to "a sequence of *people*" (Ex. 2016, 94-98) contradicts with Dr. Clark's rebuttal testimony on "a sequence of *bits*" of a data item (Ex. 1092 ¶ 28). Dr. Clark in the prior deposition also testified that there are examples of sequences with intervening gaps including Fibonacci sequences, random sequences, odd sequences, and even sequences. Opp. 15 (citing Ex. 2016, 191-193). Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude the alleged inconsistent statements in Dr. Clark's rebuttal declaration.

For the foregoing reasons, we decline to exclude Dr. Clark's rebuttal declaration (Ex. 1092).

III. CONCLUSION

EMC has met its burden of proof, by a preponderance of the evidence, in showing that claims 10, 21, and 34 the '539 patent are unpatentable based on the following grounds of unpatentability:

Claim	Basis	References
10 and 21	§ 102(b)	Langer
34	§ 103(a)	Langer and Woodhill
10 and 21	§ 103(a)	Kantor
34	§ 103(a)	Kantor and Langer
10 and 21	§ 103(a)	Woodhill and Fischer

IV. ORDER

In consideration of the foregoing, it is

ORDERED that claims 10, 21, and 34 of the ’539 patent are held *unpatentable*;

FURTHER ORDERED that EMC’s Motion to Exclude Evidence is *dismissed*;

FURTHER ORDERED that PersonalWeb’s Motion to Exclude Evidence is *denied*; and

FURTHER ORDERED that, because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Peter M. Dichiara, Esq.
David L. Cavanaugh, Esq.
WILMER CUTLER PICKERING HALE & DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com

PATENT OWNER:

Joseph A. Rhoa, Esq.
Updeep. S. Gill, Esq.
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

Trials@uspto.gov
571-272-7822

Paper 66
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION,
Petitioner,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00086
Patent 7,949,662 B2

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

TURNER, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

EMC Corporation ("EMC") filed a petition on December 17, 2012, requesting an *inter partes* review of claim 30 of U.S. Patent No. 7,949,662 B2 ("the '662 Patent"). Paper 3 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") filed a patent owner preliminary response. Paper 9 ("Prelim. Resp."). Taking into account the patent owner preliminary response, the Board determined that the information presented in the petition demonstrated that there was a reasonable likelihood that EMC would prevail with respect to claim 30. Pursuant to 35 U.S.C. § 314, the Board instituted this trial on May 17, 2013, as to claim 30 of the '662 Patent. Paper 14 ("Dec.").

After institution, PersonalWeb filed a patent owner response (Paper 33 ("PO Resp.")), and EMC filed a reply to the patent owner response (Paper 41 ("Reply")). Oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is entered pursuant to 35 U.S.C. § 318(a). We hold that claim 30 of the '662 Patent is unpatentable under 35 U.S.C. § 103.

[1] This proceeding, as well as IPR2013-00082, IPR2013-00083, IPR2013-00084, IPR2013-00085, and IPR2013-00087, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 65.

A. Related Proceeding

EMC indicates that the ’662 Patent is the subject of litigation titled *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED (E.D. Tex.). Pet. 1.

B. The ’662 Patent

The ’662 Patent relates to a data processing system that identifies data items using substantially unique identifiers, otherwise referred to as True Names, which depend on all the data in the data item and only on the data in the data item. Ex. 1001, 1:17-21, 3:27-30, 5:66-6:6. According to the ’662 Patent, the identity of a data item depends only on the data and is independent of the data item’s name, origin, location, address, or other information not directly derivable from the data associated therewith. *Id*. at 3:34-37. The ’662 Patent also examines the identities of a plurality of data items in order to determine whether a particular data item is present in the data processing system. *Id*. at 3:38-43.

The’662 Patent further discloses accessing data items by referencing their identities or True Names independent of their present location in the data processing system. *Id*. at 33:46-48. The actual data item or True file corresponding to a given data identifier or True Name is capable of residing anywhere on the data processing system, *i.e.*, locally, remotely, offline, etc. *Id*. at 33:46-48. If a requested data item or True File is local with respect to the data processing system, a prospective user can access the data in the True File. *Id*. at 33:48-50. If a requested data item or True File is not local with respect to the data processing system, a prospective user may use the True File registry to determine the location of copies of the True File according to its given True Name. *Id*. at 33:50-54. However, if for some reason a prospective user cannot locate a copy of the requested data item or True File, the processor employed by the user may invoke the Request True File remote mechanism to submit a general request for the data item or True File to all the processors in the data processing system. *Id*. at 34:58-64.

C. Challenged Claim

Claim 30 recites the following (emphasis added):

> 30. A computer-implemented deletion method operable in a file system comprising (i) a plurality of servers; (ii) a list indicating, for each of a plurality of files in the file system, a corresponding status,
>
> wherein, for each of a plurality of data items in the file system, said data items each consisting of a corresponding sequence of one or more parts; and
>
> wherein each data item has a corresponding digital data item identifier, said digital data item identifier for the data item being based, at least in part, on the contents of the data item, wherein two identical data items in the file system have the same digital data item identifier; and
>
> wherein each part is replicated on multiple servers of said plurality of servers; and
>
> wherein said list includes digital data item identifiers for data items for which changes are to be made in the file system,
>
> the method comprising the steps of:
>
> (A) obtaining a particular digital data item identifier of a particular data item, *said particular digital data item identifier*

> *of said particular data item being obtained in response to an attempt to delete said particular data item in said file system*;
>
> (B) *updating a record in said list to reflect deletion of said particular data item from the file system*, said record including the particular digital data item identifier to the list.

Ex. 1001, 43:28-55 (emphasis added).

D. Prior Art Relied Upon

EMC relies upon the following prior art references:

Frederick W. Kantor, "*FWKCS (TM) Contents-Signature System Version 1.22*," FWKCS122.REF (Aug. 10, 1993) (Ex. 1004, hereinafter "Kantor").

Mahadev Satyanarayanan et al., "*Coda: A Highly Available File System for a Distributed Workstation Environment*," IEEE Transactions on Computers, Vol. 39, No. 4 (April 1990) (Ex. 1026, hereinafter "Satyanarayanan").

E. Ground of Unpatentability

The Board instituted the instant trial based on the following ground of unpatentability:

Claim	Basis	References
30	§ 103(a)	Kantor and Satyanarayanan

II. ANALYSIS

A. Claim Construction

We begin our analysis by determining the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). An inventor may rebut that presumption by providing a definition of the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). In the absence of such a definition, limitations are not to be read from the specification into the claims. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993).

In the Decision on Institution, we construed the claim term "data item" to mean "sequence of bits," and observed that in the context of the specification, the meaning also includes one of the following: (1) the contents of a file; (2) a portion of a file; (3) a page in memory; (4) an object in an object-oriented program; (5) a digital message; (6) a digital scanned image; (7) a part of a video or audio signal; (8) a directory; (9) a record in a database; (10) a location in memory or on a physical device or the like; and (11) any other entity which can be represented by a sequence of bits. Dec. 10. The parties agree with that claim construction. Pet. 6-7; PO Resp.

1-2. As noted in the Decision on Institution, that claim construction is consistent with the specification. Dec. 9-10 (citing Ex. 1001, 1:56-57 ("the terms 'data' and 'data item' as used herein refer to sequences of bits."); *id*. at 1:56-61, 1:66–2:4). We discern no reason to deviate from that claim construction for the purposes of this decision.

B. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc*., 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). In that regard, an obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418; *see also Translogic*, 504 F.3d at 1259.

We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art."

Paulsen, 30 F.3d at 1480. Moreover, "it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." *In re Preda*, 401 F.2d 825, 826 (CCPA 1968). We analyze the instituted ground of unpatentability in accordance with the above-stated principles.

C. Claim 30 – Obviounesss over Kantor and Satyanarayanan

EMC asserts that claim 30 is unpatentable under 35 U.S.C. § 103(a) as obvious over Kantor and Satyanarayanan. Pet. 38-47. As support, EMC provides detailed explanations as to how each claim element, arranged as recited in the claim, is disclosed by Kantor, Satyanarayanan and/or the combination of both. *Id*. Additionally, EMC also directs our attention to the declaration of Dr. Clark. *Id*. (citing Ex. 1009).

PersonalWeb counters that Kantor does not disclose obtaining a digital data item identifier in response to an attempt to delete, that Kantor's "d" flag is not a "status" indicator, and that it would not have been obvious to have modified Kantor to combine the MULTIS list with the deleted.log, as argued by EMC. PO Resp. 3-12. PersonalWeb also alleges that Kantor is not a "printed publication" within the meaning of 35 U.S.C. § 102(b). *Id*. at 13-19. In support of its argument, PersonalWeb proffers Mr. Todd Thompson's declaration (Ex. 2014).

Upon review of the parties' arguments and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claim 30 is unpatentable under 35 U.S.C. § 103(a) as being obvious over

Kantor and Satyanarayanan. We also determine that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

Kantor

Kantor describes a method of identifying duplicate files. Ex. 1004, 2-4, 48-49. In particular, Kantor applies a hash function (*e.g.*, a cyclic residue check or cyclic redundancy check (CRC)) to each file within a zip file to obtain the contents signature for each file. *Id.* at 6-8, 48-49. Each contents signature is a string of bits generated from the contents of a file. *Id.*

For each zip file, Kantor creates zip-file contents signatures by hashing the contents signatures for the files contained within the zip file ("a hash of hashes"). *Id.* at 2, 9. As described by Kantor, this is done by "adding together all the 32_bit CRC's for the files in the zip file, modulo 2^32, separately adding together their uncompressed file_lengths modulo 2^32, and then arranging the two resulting hexadecimal numbers as a single structure." *Id.* at 9. Dr. Clark testifies that addition modulo 2^32 is another well-known simple hashing function that uses addition to calculate a value for a file based on the file's contents. Ex. 1009 ¶ 20. Kantor further compares the zip-file contents signatures to check for duplicate files. Ex. 1004, 2 of Preface, 5, 9.

According to Kantor, contents signatures and zip-file contents signatures are useful to identify files that have the same contents stored on the electronic bulletin board systems. Ex. 1004, 2 of Preface, 5, 9. For example, when uploading a zip file, the system determines whether that zip file already exists in the system using the zip-file contents signature, and then determines whether the inner files of that zip file already exist in the system using the contents signatures for the inner files. *Id*. at 9.

EMC has acknowledged that Kantor fails to disclose the underlying storage system of the BBS, and, thus, does not disclose that files are replicated on multiple servers, per claim 30. Pet. 47. Satyanarayanan discloses a network-based file replication system, where copies of files are stored at multiple servers (Ex. 1026, Abstract). EMC also argues that a person of ordinary skill would have found it obvious to modify Kantor to meet that limitation in view of Satyanarayanan. Pet. 43-44. On this record, we concur with the analysis of Dr. Clark, that it would have been obvious to combine Kantor and Satyanarayanan to provide more reliable storage systems for the BBS's files (Ex. 1009 ¶ 47).

Obtaining a digital data item identifier in response to an attempt to delete

PersonalWeb argues that Kantor alone is relied upon to teach the step of claim 30 of "obtaining a particular digital data item identifier of a particular data item, *said particular digital data item identifier of said particular data item being obtained in response to an attempt to delete said particular data item in said file system;*" (emphasis added). PO Resp. 3. PersonalWeb argues that Kantor fails to teach this element because Kantor fails to disclose obtaining a digital data item identifier *in response to an attempt to delete* a particular data item. *Id*. at 4. PersonalWeb continues that the MULTIS list is formulated *prior* to any attempt to delete a file, and is used by the user to add a "d" to a specific column therein, and, subsequently, fwkc17d is run to delete the marked files. *Id*. at 4-5.

In its Reply, EMC contends that PersonalWeb and its experts acknowledge that the system in Kantor generates and maintains a master list of the contents signatures called CSLIST.SRT, and the MULTIS feature is used to analyze the CSLIST, and identify and list the files for which multiple copies exist. Reply 2. EMC also argues that PersonalWeb is mistaken in asserting that it is only when the fwkc17d command is run that "an attempt to delete" begins. *Id*. at 3. In contrast, EMC argues that it is clear from Kantor that the attempt to delete begins with the MULTIS command. *Id*.; Ex. 1004, 189.

We are persuaded that EMC has demonstrated that its view of Kantor is correct. Kantor describes the process of running FWKCS as doing "a partial clean up" and that the running of the MULTIS command "put[s] all of the duplicate zipfiles together in groups, in the file MULTIS." Ex. 1004, 189. Thereafter, a word processor is used to add a "d" to the line of the MULTIS file for the files to be deleted. *Id*. The contents signatures of the listed files are "obtained" in response to the "attempt to delete." *Id*. Thus, the process begins with the MULTIS command, which obtains file information, including contents signatures, from the CSLIST. We are persuaded that this is equivalent to the process step of "obtaining" in claim 30.

In addition, as EMC notes, the ’662 Patent discloses that when an attempt to delete a file occurs, the system obtains the file’s True Name from the Local Directory Extensions table and copies it to an audit file. Reply 4; Ex. 1001, 21:51-22:6. Thus, we agree with EMC that “obtaining” step of claim 30 must include within its meaning the copying of a contents signature to a file, per the description of deleting a file in the ’662 Patent. Thus, even if the deletion process in Kantor begins with the running of the fwkc17d command, as argued by PersonalWeb (PO Resp. 3-6), we agree that the digital data item identifier is obtained in response to an attempt to delete. Thus, we determine that EMC has demonstrated sufficiently that Kantor obtains a digital data item identifier in response to an attempt to delete, per claim 30.

In addition, PersonalWeb also argues that the “Exclude” feature of Kantor, which was relied upon by EMC in its petition (Pet. 24-25), cannot disclose the obtaining step of claim 30. PO Resp. 6-7. EMC counters that marking of an entry with an “x” begins the attempt to exclude the file, similar to the arguments discussed above with respect to the use of the MULTIS list to delete a file in Kantor. Reply 5. However, given our determination that EMC has demonstrated by a preponderance of the evidence that Kantor obtains a digital data item identifier in response to an attempt to delete, we need not reach whether the “exclude” feature also provides the same functionality.

Kantor's "d" flag

PersonalWeb argues that the "d" placed by a file in the MULTIS list does not indicate a "status," per claim 30, as it does not indicate that the file has been deleted, but simply indicates that a user may want to delete the file. PO Resp. 7-8. EMC counters that the "d" placed by a file indicator are plainly status indicators, and cites Dr. Clark's testimony that a person of ordinary skill in the art would have understood that a "flag" is a status indicator. Reply 6; Ex. 1083 ¶ 19.

We agree with EMC that the "d" is an indicator and must provide a "status." Claim 30 provides, in part, "a list indicating, for each of a plurality of files in the file system, a corresponding status." While PersonalWeb argues that "[a] user's desire is not the 'status' of a file" (PO Resp. 7), PersonalWeb fails to point to a definition of status that would so limit the term. Claim 30 does not specify what the indicator provides a status of and, as a consequence, the scope and breadth of claim 30 encompasses indicators that provide any type of status. The presence of a "d" would indicate whether a file is to be deleted or has been deleted, which would be a "status" of the file. Thus, we determine that EMC has demonstrated sufficiently that Kantor's "d" indicator is a status indicator, per claim 30.

Modifying Kantor to combine the MULTIS list with the deleted.log

PersonalWeb argues that Kantor fail to teach to suggest one aspect of claim 30, namely "updating a record in said list to reflect deletion of said particular data item from the file system." PO Resp. 8. PersonalWeb argues that the MULTIS list is not updated to reflect deletion of a record, and that it would not have been obvious to modify Kantor to combine the MULTIS list with the delete.log list to show the contents of both in the same file. *Id*. at 9. PersonalWeb argues that such a combination would be illogical and is not motivated by the disclosure of Kantor because total listing of both is already in the MULTIS list. *Id*. PersonalWeb also argues that adding entries of deleted files to the MULTIS list would make it harder to identify files with redundant signatures. *Id*. at 10.

In its Reply, EMC contends that the two files, the MULTIS list and the delete.log list, are not redundant because the delete.log list presents a chronological sequence of actual deletions and provides a complete history of operations, whereas the MULTIS list reflects the files currently in the system that are to be deleted or which were recently deleted. Reply 10. EMC also points to the testimony of Dr. Clark that explains that combining the files is both logical and obvious as a mere design choice, providing that the files could have been combined in a way that made it clear which portions of the single file comprised the original contents of the two files. *Id*.; Ex. 1083, ¶ 25.

Based on the arguments and evidence presented, we agree with EMC that it would have been obvious to have combined the lists into a single list. PersonalWeb's view of the combination is akin to a bodily incorporation of one list into another. PO Resp. 8-12. However, "[t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference . . . Rather, the test is what the combined teachings of those references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981). We see no reason for the list to be combined into a single file and to have all of the entries be intermingled. Rather, the modification of Kantor could be accomplished by combining the contents of the files into a single list while still maintaining their apparent separation. Such a combination would not be "illogical" and would not cause the confusion PersonalWeb ascribes to the modification of Kantor. As such, we are persuaded that EMC has demonstrated that combining Kantor's MULTIS list and the delete.log list into a single file would have taught the updating of a record in a list to reflect deletion of the particular data item from the file system, per claim 30.

Evidence of non-obviousness

PersonalWeb further submits that its evidence of non-obviousness rebuts EMC's evidence of obviousness. PO Resp. 12-13. In support of its argument, PersonalWeb directs our attention to three licensing agreements, as well as the declaration of Mr. Kevin Bermeister. *Id*. at 12 (citing Exs. 2010-12; Ex. 2009 ¶¶ 3-9). PersonalWeb argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between claim 30 of the '662 Patent and the above-identified license agreements. Reply 11-12. EMC argues that each of the licenses granted rights to more than just claim 30, and involved related parties with interlocking ownership and business interests. *Id.* We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and claim 30.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *In re GPAC Inc.*, 57 F.3d 1573, 1580 (Fed. Cir. 1995). Further, without a showing of nexus, "the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness" when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000)*; see Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus. Neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010-12) are directed to the claimed subject matter recited in claim 30. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in claim 30. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Furthermore, even if we assume that above-identified licenses establish some degree of industry respect for the claimed subject matter recited in claim 30, that success is outweighed by the strong evidence of obviousness over Kantor and Satyanarayanan discussed above.

Based on the record before us, including the evidence of obviousness presented by EMC and the evidence of secondary considerations regarding licensing activities presented by PersonalWeb, we conclude that EMC has demonstrated by a preponderance of the evidence, that claim 30 would have been obvious over the combination of Kantor and Satyanarayanan.

Whether Kantor is a "printed publication"

In its petition, EMC takes the position that Kantor is a "printed publication" under 35 U.S.C. § 102(b). Pet. 4-5. EMC asserts that Kantor has been publicly available since August 1993, which is prior to the critical date, April 11, 1995, one year before the earliest priority date claimed by the '662 Patent. *Id*. To substantiate its position, EMC explains that Kantor is "a published manual that describes a software program called the Frederick W. Kantor Contents-Signature System Version 1.22 ('FWKCS')." *Id*. at 38 (citing Ex. 1004, Title Page). EMC maintains that Dr. Frederick Kantor distributed Kantor—the user manual (version 1.22), the version relied upon by EMC (*see* Ex. 1004)—with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where Kantor could be downloaded by anyone. Pet. 4-5, n. 2 (citing Ex. 1004, 3, 158-59). According to EMC, Kantor was accessible to others in the relevant community of the users and system operators of electronic Bulletin Board Systems. *Id.* In support of its position, EMC proffers a declaration of Mr. Michael A. Sussell (Ex. 1041) and declarations of Mr. Jason S. Sadofsky (Ex. 1072; Ex. 1082).

In its patent owner response, PersonalWeb counters that Kantor is not a "printed publication." PO Resp. 13-19. In particular, PersonalWeb alleges that EMC has not established that the specific version of Kantor existed prior to the critical date. *Id*. at 14. PersonalWeb contends that there is no evidence that Kantor was disseminated publicly, catalogued, or indexed in a meaningful way. *Id*. at 14-15. It is PersonalWeb's view that EMC fails to establish that one with ordinary skill in the art, exercising reasonable diligence, would have located Kantor prior to the critical date. *Id*. at 13.

We have reviewed the parties' arguments and supporting evidence. Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. Rather, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

The determination of whether a given reference qualifies as a prior art "printed publication" involves a case-by-case inquiry into the facts and circumstances surrounding the reference's disclosure to members of the public. *In re Klopfenstein*, 380 F.3d 1345, 1350 (Fed. Cir. 2004). The key inquiry is whether the reference was made "sufficiently accessible to the public interested in the art" before the critical date. *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989); *In re Wyer*, 655 F.2d 221, 226 (CCPA 1981).

“A given reference is ‘publicly accessible’ upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it.” *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006).

Indexing is not “a necessary condition for a reference to be publicly accessible,” but it is only one among many factors that may bear on public accessibility. *In re Lister*, 583 F.3d 1307, 1312 (Fed. Cir. 2009). In that regard, “while often relevant to public accessibility, evidence of indexing is not an absolute prerequisite to establishing online references . . . as printed publications within the prior art.” *Voter Verified, Inc. v. Premier Election Solutions, Inc.*, 698 F.3d 1374, 1380 (Fed. Cir. 2012).

Contrary to PersonalWeb’s assertion that Kantor did not exist prior to the critical date and there is no evidence that Kantor was disseminated publicly, Kantor itself shows a copyright date of “1988-1993” and a posted date of “1993 August 10.” Ex. 1004, Title Page, the first page after the Title Page (“All of the programs and documents, comprising the entire contents of this Authenticity Verification Zip file FWKCS122.ZIP, together with this Zipfile itself, are, in accordance with their respective dates of creation or revision, (C) Copyright Frederick W. Kantor 1988-1993.”). Kantor also states:

> The FWKCS(TM) Contents_Signature System has become a robust platform for supporting contents_signature functions. FWKCS provides many functions and options for application in a public, commercial, school, institutional, or governmental

> environment. Extensive technical support is of special value in helping such users to benefit more fully from these many features.
>
> Registered FWKCS hobby BBS users are able to receive a modest amount of assistance, and are invited to participate in the FWKCS conference on The Invention Factory BBS, echoed via Execnet.
>
> Commercial, school, institutional, and governmental users, with their special support needs, are invited to discuss terms for obtaining such assistance.
>
> . . . .
>
> To get a new version of FWKCS, download FWKCSnnn.ZIP from The Invention Factory BBS, where nnn is the new version number without a decimal point. These special downloads are available at no fee, from a 43_line hunt_up group of USR Dual Standard modems, at 2400-16800 bits/sec (including V32.bis).

Ex. 1004, 158-159. It is clear from Kantor that, during the 1988-1993 timeframe, Dr. Kantor had posted many versions of his software and user manual—including Kantor (version 1.22),, the version relied upon by EMC (Ex. 1004)—on electronic Bulletin Board Systems.

Mr. Sussell, the co-owner and system operator of the Invention Factory Bulletin Board System, testifies that the Invention Factory Bulletin Board System is a computer system that allows users to share files, messages, and articles, as well as search, upload, and download files. Ex. 1041 ¶¶ 3-4. According to Mr. Sussell, he and his wife launched the Invention Factory Bulletin Board System in 1983, and it had over 3,000 subscribers by mid-1993. *Id.* ¶ 6. Mr. Sussell testifies that, by 1993, the system provided all users keyword search functionality and access to various descriptive and meaningful directories. *Id.* ¶¶ 8-10.

More importantly, Mr. Sussell testifies that the Invention Factory Bulletin Board System "extensively utilized and hosted current versions of FWKCS software on its [Bulletin Board System]," and "made publicly accessible and available the complete FWKSC ZIP file that contained both the software as well as related documentation such as user manuals" prior to the critical date. *Id.* ¶ 15; *see id.* ¶¶ 16-27. Specifically, Mr. Sussell testifies that users would have found Kantor by performing keyword searches on the Invention Factory Bulletin Board System. *Id.* ¶ 21. Mr. Sussell also indicates that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about Dr. Kantor's software on the "Welcome" screen, and made the FWKCS Zip file available in four different directories. *Id.* ¶¶ 18-20. Mr. Sussell further testifies that computer disks that contain the FWKCS Zip file were distributed at various Bulletin Board System conferences. *Id.* ¶ 18.

Mr. Sadofsky, a technology archivist and software historian, testifies that he personally verified the authenticity of Kantor—the user manual (version 1.22), the version relied upon by EMC (Ex. 1004)—by comparing it with a "1993 archived" version, and determined that Kantor is identical to the "1993 archived" version. Ex. 1072 ¶¶ 14-17. Mr. Sadofsky testifies that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id.* ¶ 16; Ex. 1082 ¶¶ 10-11; Ex. 2014 ¶ 5. According to Mr. Sadofsky, Kantor was publicly accessible prior to the critical date. *Id.*

PersonalWeb also asserts that Kantor was buried and hidden in the zip file in a manner such that “it would not have been located and accessed by persons interested and ordinarily skilled in the art exercising reasonable diligence even if they had access to the ZIP file.” PO Resp. at 17-18 (citing Ex. 2014). However, PersonalWeb’s supporting evidence, Mr. Thompson’s declaration (Ex. 2014), does not substantiate PersonalWeb’s assertion. Upon review of Mr. Thompson’s declaration, we observe that Mr. Thompson downloaded the FWKCS Zip file—the zip file that contains the software and Kantor, the user manual—without any difficultly. Ex. 2014 ¶ 5. Significantly, Mr. Thompson did not follow the instructions provided with the zip file, nor did he use the appropriate computer environment (DOS 3.0 or an IBM OS/2 2.0) that was used normally in 1993-1994 timeframe, but instead he used non-compatible software (DOS 8.0 and 32-bit Windows XP operating system that was released in 2001). Ex. 2014 ¶¶ 6-11; Ex. 1082 ¶¶ 5, 14. Once he followed the instructions and unzipped the FWKCS Zip file, Mr. Thompson located Kantor without difficulty. Ex. 2014 ¶¶ 20-22.

Mr. Sadofsky confirms that the README.TXT file provides simple instructions and, if a user follows the instructions and uses the operating system that was used normally in 1993-1994 timeframe, the user could locate Kantor without difficulty. Ex. 1082 ¶¶ 13-17. In fact, Mr. Sadofsky demonstrated, in his declaration, several relatively easy ways for a user to access Kantor—with or without installing the software, and with or without help screens. Ex. 1082 ¶¶ 8-16 (II. README.TXT); ¶¶ 17-20 (III. GETLOOK.BAT); ¶¶ 21-22 (IV. FWKCS122 Start Screen and In-

Program Help). Based on the evidence before us, we determine that Kantor was available to the extent that persons interested and ordinarily skilled in the art, exercising reasonable diligence, could locate it.

PersonalWeb's argument that EMC's witnesses personally did not post or review Kantor prior to the critical date also is unavailing. PO Resp. 14-16 (citing Ex. 2015, 52-55; Ex. 2013, 29-30; Ex. 2016, 98). It is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986) (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible"); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'"); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would know of the invention.).

The evidence on this record clearly support that Kantor was posted on a publicly accessible site—the Invention Factory Bulletin Board System—well known to those interested in the art, and could be downloaded and retrieved from that site, and, therefore, Kantor, an electronic publication, is considered a "printed publication" within the meaning of 35 U.S.C.

§ 102(b). *See Wyer*, 655 F.2d at 226 (An electronic publication, including an on-line database or Internet publication, is considered to be a "printed publication" "upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it and recognize and comprehend therefrom the essentials of the claimed invention without need of further research or experimentation.").

For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b). Therefore, EMC may rely upon Kantor for its asserted ground of unpatentability under 35 U.S.C. § 103(a).

D. EMC's Motion to Exclude

EMC seeks to exclude the following exhibits: (1) three license agreements (Exs. 2010-12); (2) Mr. Bermeister's declarations (Exs. 2009, 2018) relating to those license agreements; and (3) Mr. Thompson's declaration (Ex. 2014). Paper 50 ("Pet. Mot."). PersonalWeb filed the license agreements and Mr. Bermeister's declarations as evidence of non-obviousness to rebut EMC's assertion that claim 30 would have been obvious over the combination of Kantor and Satyanarayanan. PO Resp. 12-13. As to Mr. Thompson's declaration, PersonalWeb proffered that evidence to support its assertion that Kantor—a user manual that was disseminated publicly with the software in a zip file—was not made sufficiently accessible to a person interested and ordinarily skilled in the art. *Id*. at 16-18. PersonalWeb opposes EMC's motion to exclude. Paper 56. In response, EMC filed a reply to PersonalWeb's opposition to its motion to exclude. Paper 59.

With respect to the license agreements and Mr. Bermeister's declarations (Exs. 2009-2012, 2018), EMC argues that they are irrelevant under Federal Rule of Evidence 402[2], highly prejudicial, confusing, and misleading under Federal Rule of Evidence 403. *Id*. at 8-13. As to Mr. Thompson's declaration, EMC argues that it should be excluded under Federal Rule of Evidence 402. *Id*. at 14-15. Specifically, EMC alleges that: (1) Mr. Thompson does not possess the skill of a person of ordinary skill in the art (*id*. at 14-15 (citing Ex. 1077, 13-14)); (2) Mr. Thompson did not use compatible software from the relevant time period (*id*. at 15 (citing Ex. 1077, 40-41; Ex. 2014, 4, 6)); and (3) Mr. Thompson did not follow the instructions provided with the zip file (*id*. at 15 (citing Ex. 1077, 32-35)).

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding PersonalWeb's supporting evidence, we have determined that Kantor is a "printed publication" under 35 U.S.C. § 102(b), and EMC has demonstrated, by a preponderance of the evidence, that claim 30 is unpatentable over the combination of Kantor and Satyanarayanan.

[2] As stated in 37 C.F.R. § 42.62, the Federal Rules of Evidence generally apply to proceedings, including *inter partes* reviews.

Accordingly, EMC's motion to exclude evidence is *dismissed* as moot.

E. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following items of evidence: (1) Kantor (Ex. 1004); (2) certain documents (Exs. 1038-1040, 1043-1046, 1065, 1066, 1074-1076) and the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) regarding those documents; (3) the declarations of Messrs. Sussell and Sadofsky regarding Kantor (Exs. 1041, 1072, 1082) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66); and (4) Clark's rebuttal declaration (Ex. 1083 ¶¶ 26-27, 30). Paper 50 ("PO Mot.").

EMC opposes PersonalWeb's motion to exclude. Paper 57 ("Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 60 ("PO Reply"). For the reasons stated below, PersonalWeb's motion to exclude is *denied*.

Kantor

PersonalWeb alleges that Kantor should be excluded as unauthenticated and inadmissible hearsay under Federal Rules of Evidence 901 and 902. PO Mot. 1, 6. In particular, PersonalWeb argues that "[n]o witness of record has personal knowledge of Kantor existing prior to [the critical date], and electronic data such as Kantor is inherently untrustworthy because it can be manipulated from virtually any location at any time." *Id.*

at 2-4. According to PersonalWeb, the dates provided by Kantor are inadmissible hearsay because Kantor is not self-authenticating. *Id*. at 2, 5-6.

EMC argues that Kantor has been authenticated under Federal Rules of Evidence 901, and that the document is not hearsay, because it is being offered for what it describes—not for the truth of its disclosures. Opp. 1-10. In particular, EMC disagrees with PersonalWeb that Kantor cannot be authenticated without direct testimony from a witness with personal knowledge that Kantor existed prior to the critical date. Opp. 1. EMC asserts that it need "only produce evidence 'sufficient to support a finding' that the reference 'is what the proponent claims it is.'" *Id*. at 1-2 (citing Fed. R. Evid. 901(a)). EMC also contends that testimony from Messrs. Sussell and Sadofsky provides sufficient evidence to authenticate Kantor. Opp. 1-5 (citing Exs. 1041, 1072, 1082).

In its reply, PersonalWeb argues that Federal Rules of Evidence identified by EMC are not applicable to Kantor, because Mr. Sussell did not post or review Kantor prior to critical date. PO Reply 1-5 (citing Ex. 2015, 32-36, 55, 55, 65). PersonalWeb also alleges that Kantor's authenticity is suspicious, as electronic data are inherently untrustworthy and there is no chain of custody. *Id*.

We have considered PersonalWeb's arguments as well as EMC's contentions and supporting evidence. We are not persuaded that Kantor should be excluded.

At the outset, we disagree with PersonalWeb's position that a witness cannot authenticate a document, unless the witness is the author of the document or the witness has reviewed the document prior to the critical date. Federal Rule of Evidence 901(a) states that the authentication requirement is satisfied if the proponent presents "evidence sufficient to support a finding that the item is what the proponent claims it is." Therefore, neither a declaration from the author, nor evidence of someone actually viewing the document *prior to critical date*, is required to support a finding that the document is what it claims to be. *See also Hall*, 781 F.2d at 899 (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible."); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'").

Further, it is well settled that an uninterrupted chain of custody is not a prerequisite to admissibility, but rather gaps in the chain go to weight of the evidence. *U.S. v. Wheeler*, 800 F.2d 100, 106 (7th Cir. 1986); *see also U.S. v. Aviles*, 623 F.2d 1192, 1198 (7th Cir. 1980) ("If the trial judge is satisfied that in reasonable probability the evidence has not been altered in any material respect, he may permit its introduction.") (Citation omitted). There is a strong public policy for making all information filed in a quasi-judicial administrative proceeding available to the public, especially in an *inter partes* review, which determines the patentability of a claim in an issued patent. It is within the Board's discretion to assign the appropriate weight to be accorded to evidence.

Although Messrs. Sussell and Sadofsky personally did not post or review the particular version of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—prior to the critical date, they have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. *See Hall*, 781 F.2d at 899; *Wyer*, 655 F.2d at 226; *Bayer*, 568 F.2d at 1361.

Notably, Mr. Sussell, the co-founder and system operator of the Invention Factory Bulletin Board System, testifies that Dr. Kantor released the first version of his software on the Invention Factory Bulletin Board System in the 1980s, and the system continuously utilized and hosted current versions of the software and user manuals. Ex. 1041 ¶¶ 3, 13, 15. Mr. Sussell also testifies that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about Dr. Kantor's software on the "Welcome" screen, and made FWKCS Zip file—a zip file that contains both the software and user manual—publicly accessible and available under four different directories. *Id.* ¶ 18. According to Mr. Sussell, the Invention Factory Bulletin Board System had over 3,000 subscribers, in the 1993 timeframe, and all of the users had the capability to perform keyword searches to retrieve FWKCS Zip file. *Id.* ¶¶ 6, 21.

Although we are cognizant that electronic documents downloaded from websites normally are not self-authenticating, it has been recognized that "[t]o authenticate printouts from a website, the party proffering the evidence must produce some statement or affidavit from someone with knowledge of the website . . . for example a web master or someone else with personal knowledge would be sufficient." *St. Luke's Cataract and Laser Institute v. Sanderson*, 2006 WL 1320242, *2 (M.D. Fla. 2006) (citing *In re Homestore.com, Inc. Sec.Litig.*, 347 F. Supp. 2d 769, 782 (C.D. Cal. 2004)) (quotation marks omitted); Ex. 2024; *see also Market-Alerts Pty. Ltd. v. Bloomberg Finance L.P.*, 922 F. Supp. 2d 486, 493, n.12 (D. Del. 2013) (citing *Keystone Retaining Wall Sys., Inc. v. Basalite Concrete Prods., LLC*, 2011 WL 6436210, at *9 n.9 (D. Minn. 2011)) (documents generated by a website called the Wayback Machine have been accepted generally as evidence of prior art in the patent context); *U.S. v. Bansal*, 663 F.3d 634, 667-68 (Fed. Cir. 2011) (concluding that the screenshot images from the Internet Archive were authenticated sufficiently under Federal Rule of Evidence 901(b)(1) by a witness with personal knowledge of its contents, verifying that the screenshot the party seeks to admit are true and accurate copies of Internet Archive's records).

Here, Mr. Sadofsky, who is a technology archivist and software historian and currently is an archivist for the Internet Archive, testifies that he launched the website textfiles.com and a subdomain cd.textfiles.com to collect software, data files, and related materials from Bulletin Board Systems. Ex. 1072 ¶¶ 9-11. According to Mr. Sadofsky, textfiles.com and cd.textfiles.com are dedicated to preserving, archiving, and providing free access to unaltered historical software programs and information that initially were made available on the Bulletin Board System. *Id.* Mr. Sadofsky states that he previously archived the FWKCS Zip file (FWKCS122.ZIP) that contains Dr. Kantor's software and user manual to cd.textfiles.com from his own copy of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM. *Id.* ¶ 14 (citing Ex. 1048). Mr. Sadofsky also testifies that he personally verified the authenticity of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—by comparing it with the "1993 archived" version and determined that Kantor is identical to the "1993 archived" version. Ex. 1072 ¶¶ 13-15. Mr. Sadofsky confirms that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id.* ¶ 16; Ex. 1082 ¶¶ 10-11; Ex. 2014 ¶ 5. Mr. Sadofsky concludes that Kantor was publicly accessible prior to the critical date. Ex. 1072 ¶¶ 13, 16.

Moreover, we agree with EMC that Kantor also has been authenticated as an "ancient document" under Federal Rule of Evidence 901(b)(8).[3] Opp. 7 Kantor is "at least 20 years old and can be found in . . . an October 1993 *Simtel* CD-ROM – a place where an authentic 20-year old document distributed through a [Bulletin Board System] would likely be." *Id.*; Ex. 1072 ¶¶ 7-8; *see also* Fed. R. Evid. 901(b)(8) 2012 Adv. Comm.

[3] Fed. R. Evid. 901(b)(8). Evidence About Ancient Documents or Data Compilations. For a document or data compilation, evidence that it:
(A) is in a condition that creates no suspicion about its authenticity;
(B) was in a place where, if authentic, it would likely be; and
(C) is at least 20 years old when offered.

Note ("The familiar ancient document rule of the common law is extended to include data stored electronically or by other similar means."). Moreover, testimony of Messrs. Sussell and Sadofsky has established sufficiently that Kantor is in a condition that creates no suspicion about its authenticity. Exs. 1041, 1072, 1082.

PersonalWeb does not present sufficient or credible evidence to the contrary. Based on the evidence before us, we determine that Kantor has been authenticated under Federal Rules of Evidence 901(b)(1), (b)(3), (b)(4), and (b)(8) to warrant its admissibility.

PersonalWeb's hearsay argument regarding Kantor also is unavailing. As EMC notes (Opp. 8), a "prior art document submitted as a 'printed publication' under 35 U.S.C. § 102(a) is offered simply as evidence of what it described, not for proving the truth of the matters addressed in the document." *See, e.g.*, *Joy Techs., Inc. v. Manbeck*, 751 F. Supp. 225, 233 n.2 (D.D.C. 1990), *judgment aff'd*, 959 F.2d 226 (Fed. Cir. 1992); Fed. R. Evid. 801(c) 1997 Adv. Comm. Note ("If the significance of an offered statement lies solely in the fact that it was made, no issue is raised as to the truth of anything asserted, and the statement is not hearsay."). Therefore, Kantor is not hearsay under Federal Rule of Evidence 801(c).

We further agree with EMC that the posted date of "1993 August 10" or the copyright date of "1988-1993" on the Title page of Kantor is not a basis for excluding Kantor, as testimony from Messrs. Sussell and Sadofsky sufficiently establishes that Kantor existed as of August 10, 1993, prior to the critical date. Opp. 8. More importantly, the computer-generated timestamp—August 10, 1993, at 1:22 AM—of the "1993 archived" version of Kantor (Ex. 1072 ¶¶ 14-15; Ex. 1082 ¶¶ 10-11; Ex. 2014 ¶ 5) also independently corroborates Kantor's existence as of August 10, 1993. *See, e.g., U.S. v. Khorozian*, 333 F.3d 498, 506 (Fed. Cir. 2003) (concluding that an automatically generated time stamp on a fax was not a hearsay statement because it was not uttered by a person). Accordingly we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Kantor as impermissible hearsay.

For the foregoing reasons, we decline to exclude Kantor.

Documents Corroborating Witnesses' Knowledge and Recollections

PersonalWeb asserts that certain documents submitted by EMC (Exs. 1038-1040, 1043-1046, 1065, 1066, 1074-1076) and the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) regarding those documents should be excluded because the documents have not been authenticated properly and are inadmissible hearsay. PO Mot. 6-9. PersonalWeb argues that EMC "has not established that any of these documents existed prior to the critical date, and no witness has personal knowledge of their alleged existence prior to April 11, 1995." *Id*. at 7. PersonalWeb further maintains that the documents that are Exhibits 1044, 1045, 1065, and 1066 are irrelevant, prejudicial, and confusing, as they discuss a version of Kantor different than the version relied upon by EMC (version 1.22, Ex. 1004). *Id*. at 8-9.

EMC responds that its witnesses provided those "documents to corroborate their independent knowledge and recollections." Opp. 10. EMC asserts that the documents have been authenticated under Federal Rules of Evidence 901-902 and fall within a hearsay exception under Federal Rules of Evidence 803-807. *Id.* at 10-12. We are persuaded by EMC's arguments.

As the movant, PersonalWeb has the burden of proof to establish that it is entitled to the requested relief. 37 C.F.R. § 42.20(c). As discussed previously, we disagree with PersonalWeb that documents cannot be authenticated without direct testimony from the author or a witness who actually reviewed the documents prior to the critical date. *See* Fed. R. Evid. 901(a). Significantly, PersonalWeb's motion does not contain any sufficient explanation why each document should be excluded. For instance, PersonalWeb does not explain adequately why the declaration of Mr. Sussell (Ex. 1041 ¶¶ 6, 8, 18, 27) is not sufficient to authenticate Exhibits 1043-1046, 1065, and 1066, or why the declarations of Mr. Sadofsky (Ex.1072 ¶¶ 7-17; Ex. 1087 ¶¶ 10-16) are not sufficient to authenticate Exhibits 1038-40 and 1074-1076. *See* Fed. R. Evid. 901(b)(1).[4] Nor does PersonalWeb explain sufficiently why the following documents are not self-authenticated: (1) Exhibits 1038-1040 and 1043 that include articles containing LexisNexis® trade inscriptions; (2) Exhibits 1065 and 1066 that include Usenet newsgroup periodicals containing Usenet trade inscriptions;

[4] Fed. R. Evid. 901(b)(1). Testimony of a Witness with Knowledge. Testimony that an item is what it is claimed to be.

and (3) Exhibit 1040 that contains a photograph of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM, that has Simtel trade inscriptions. *See* Fed. R. Evid. 902(6)-(7).[5]

In its motion, PersonalWeb fails to identify, specifically, the textual portions of the aforementioned exhibits that allegedly are being offered for the truth of the matter asserted, yet seeks to exclude the entirety of each exhibit. The burden should not be placed on the Board to sort through the entirety of each exhibit and determine which portion of the exhibit PersonalWeb believes to be hearsay. Rather, PersonalWeb should have identified, in its motion, the specific portions of the evidence and provided sufficient explanations as to why they constitute hearsay. Furthermore, PersonalWeb does not explain adequately why the declarations of Messrs. Sussell and Sadofsky do not provide the proper foundation and corroboration for the documents.

To the extent PersonalWeb relies upon the same arguments with respect to Kantor for excluding the documents, we have addressed those arguments above and determined that they are unavailing. We also agree with EMC

[5] Fed. R. Evid. 902. Evidence that Is Self-Authenticating
The following items of evidence are self-authenticating; they require no extrinsic evidence of authenticity in order to be admitted:

> (6) Newspapers and Periodicals. Printed material purporting to be a newspaper or periodical.
>
> (7) Trade Inscriptions and the Like. An inscription, sign, tag, or label purporting to have been affixed in the course of business and indicating origin, ownership, or control.

that the documents concerning prior versions of Kantor are relevant, and not prejudicial or confusing, as alleged by PersonalWeb, because such circumstantial evidence provides context and corroboration for the witnesses' independent knowledge and recollection.

Furthermore, we are not persuaded that the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) should be excluded. As we discuss below in the next section, Messrs. Sussell and Sadofsky have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. The documents they cite serve to corroborate their independent knowledge and recollection.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude Exhibits 1038-1040, 1043-1046, 1065, 1066, 1074-1076, as well as the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) concerning those Exhibits.

Declarations of Messrs. Sussell and Sadofsky

PersonalWeb argues that the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) should be excluded as hearsay under Federal Rule of Evidence 801, and are inadmissible under Federal Rules of Evidence 802-807 for lack of foundation and personal knowledge, and Federal Rule of Evidence 702 as improper testimony, because the witnesses personally did not review Kantor (Ex. 1004) and Simtel (Ex. 1040) prior to the critical date. PO Mot. 9-10. PersonalWeb also argues that Messrs.

Sussell and Sadofsky "are not qualified experts in the field." *Id.* at 11. PersonalWeb further alleges that Mr. Sadofsky's deposition (Ex. 2013, 30, 66) should be excluded, as it was responsive to a leading question and non-responsive to the question. *Id.*

EMC responds that the testimony of Messrs. Sussell and Sadofsky should not be excluded because their testimony is based on their own personal knowledge and recollection, and the documents they cite serve to corroborate their independent knowledge and recollection. Opp. 13. EMC further explains that the witnesses have described thoroughly the underlying facts, and, therefore, the testimony should be admitted as relevant under Federal Rules of Evidence 401-402, supported by personal knowledge and foundation under Federal Rule of Evidence 602, and proper opinion testimony under Federal Rules of Evidence 701-703. *Id.* We find EMC's contentions have merit.

PersonalWeb's arguments rest on the erroneous premise that EMC's witnesses must have reviewed Kantor or Simtel personally prior to the critical date in order to provide competent testimony regarding Kantor or Simtel. As discussed previously, it is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See, e.g., Wyer*, 655 F.2d at 226.

Although Messrs. Sussell and Sadofsky are not experts related to the claimed subject matter of the '662 patent, each witness nevertheless has sufficient personal knowledge and working experience to provide competent testimony. *See Hall*, 781 F.2d at 899. Mr. Sussell was the co-owner and system operator of the Invention Factory Bulletin Board System from 1983 to 1996. Ex. 1041 ¶ 3. Mr. Sussell's testimony is based on his personal knowledge of the relevant facts related to the Invention Factory Bulletin Board System and Kantor. *Id*. at ¶ 2. Notably, Dr. Kantor specifically thanked Mr. Sussell in his user manual for hosting Dr. Kantor's software FWKCS and for Mr. Sussell's role in its development. Ex. 1004, 3 ("To Michael Sussell, sysop of The Invention Factory (R), home board for the support of FWKCS, for bringing the problem of duplicate files to my attention and for his help in testing . . . ."); *id*. at 6 ("When Michael Sussell, sysop of The Invention Factory (R) in New York, brought to my attention the problem of duplicate files with different names, these concepts provided valuable insight into how one might proceed.").

Mr. Sadofsky is a technology archivist and software historian, and works "for the Internet Archive, a non-profit digital library offering free universal access to books, movies, and music, as well as 342 billion archived webpages available through the Wayback Machine service." Ex. 1072 ¶ 3. Mr. Sadofsky also "directed the film, *The BBS Documentary*, an eight-episode documentary about the subculture born from the creation of the [Bulletin Board System]." *Id*. at ¶ 4. Mr. Sadofsky's testimony is based on his personal knowledge of the relevant facts related to Kantor and the "1993 archived" version of Kantor. *Id*. at ¶ 2; Ex. 1087 ¶ 2. For example, Mr. Sadofsky personally verified the authenticity of Kantor by comparing it with the "1993 archived" version, and determined that Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—is identical to the "1993 archived" version. Ex. 1077 ¶¶ 14-15.

Upon review of the evidence on the record, we agree with EMC that both Messrs. Sussell and Sadofsky have disclosed sufficient underlying facts to support their testimony. For instance, the computer-generated timestamp—August 10, 1993, 1:22 AM—associated with the "1993 archived" version of Kantor corroborates their testimony regarding Kantor's existence as of August 10, 1993. Ex. 1072 ¶¶ 14-15; Ex.1082 ¶¶ 10-11; Ex. 2014 ¶ 5.

As to Mr. Sadofsky's deposition (Ex. 2013, 30, 66), PersonalWeb does not explain sufficiently why that testimony should be excluded. PO Mot. 11. Moreover, Mr. Sadofsky's deposition (Ex. 2013, 30, 66) is consistent with his direct testimony (Ex. 1072 ¶¶ 14-16), and, therefore, it would not prejudice PersonalWeb even if such evidence is not excluded.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude the declarations of Messrs. Sussell and Sadofsky (Exs. 1041, 1072, 1082) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66).

Clark's Rebuttal Declaration

PersonalWeb contends that statements in Dr. Clark's rebuttal declaration (Ex. 1083, 3:5-6, 8:5-10) contradict prior positions or arguments of EMC and should be excluded. PO Mot. 11-12. EMC counters that Dr. Clark's rebuttal declaration does not contradict earlier positions, and that Dr. Clark's positions have been the same throughout this proceeding. Opp. 13-

14.

We agree with EMC that Dr. Clark's rebuttal testimony regarding the attempt to delete begins with the generation of the MULTIS list and concludes with the running of the fwkc17d command has remained consistent throughout this proceeding. Dr. Clark testified that "Kantor further discloses a command for generating a cs-list called "MULTIS" that lists duplicate files on the system," and "[a]nother command, "FWKC17D", processes the MULTIS list to find the lines with contents-signatures that the system operator had previously marked with the "dflag" status and deletes those files." Ex. 1009 ¶ 45. In the rebuttal declaration, Dr. Clark testifies that "[i]n an attempt to delete this particular file, the sysop would generate the MULTIS file, mark the particular file with a 'd', and run the fwkc17d command to complete the deletion attempt." Ex. 1083 ¶ 12. We are not persuaded that this is an inconsistent position. Given that PersonalWeb acknowledges that EMC's petition cited to paragraph 45 of Dr. Clark's original declaration (PO Mot. 12), we cannot conclude that EMC's prior positions or arguments are inconsistent with Dr. Clark's rebuttal declaration.

For the foregoing reasons, we decline to exclude Dr. Clark's rebuttal declaration (Ex. 1083).

III. CONCLUSION

EMC has met its burden of proof, by a preponderance of the evidence, in showing that claim 30 of the ’662 Patent is unpatentable based on the following ground of unpatentability:

Claim	Basis	References
30	§ 103(a)	Kantor and Satyanarayanan

IV. ORDER

In consideration of the foregoing, it is

ORDERED that claim 30 of the ’662 Patent is held unpatentable;

FURTHER ORDERED that EMC’s Motion to Exclude Evidence is *dismissed*;

FURTHER ORDERED that PersonalWeb’s Motion to Exclude Evidence is *denied*; and

FURTHER ORDERED that, because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Peter M. Dichiara, Esq.
David L. Cavanaugh, Esq.
WILMER CUTLER PICKERING HALE & DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com

PATENT OWNER:

Joseph A. Rhoa, Esq.
Updeep S. Gill, Esq.
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

Trials@uspto.gov
571-272-7822

Paper 69
Entered: May 15, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION,
Petitioner,

v.

PERSONALWEB TECHNOLOGIES, LLC and
LEVEL 3 COMMUNICATIONS, LLC,
Patent Owners.

Case IPR2013-00087
Patent 8,001,096 B2

Before KEVIN F. TURNER, JONI Y. CHANG, and
MICHAEL R. ZECHER, *Administrative Patent Judges*.

TURNER, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

EMC Corporation ("EMC") filed a petition on December 17, 2012, requesting an *inter partes* review of claims 1, 2, 81, and 83 of U.S. Patent No. 8,001,096 B2 ("the '096 Patent"). Paper 3 ("Pet."). PersonalWeb Technologies, LLC and Level 3 Communications, LLC (collectively, "PersonalWeb") filed a patent owner preliminary response. Paper 11 ("Prelim. Resp."). Taking into account the patent owner preliminary response, the Board determined that the information presented in the petition demonstrated that there was a reasonable likelihood that EMC would prevail with respect to claims 1, 2, 81, and 83. Pursuant to 35 U.S.C. § 314, the Board instituted this trial on May 17, 2013, as to claims 1, 2, 81, and 83 of the '096 Patent. Paper 16 ("Dec.").

After institution, PersonalWeb filed a patent owner response (Paper 37 ("PO Resp.")), and EMC filed a reply to the patent owner response (Paper 44 ("Reply")). Oral hearing was held on December 16, 2013.[1]

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is entered pursuant to 35 U.S.C. § 318(a). We hold that claims 1, 2, 81, and 83 of the '096 Patent are unpatentable under 35 U.S.C. § 103.

[1] This proceeding, as well as IPR2013-00082, IPR2013-00083, IPR2013-00084, IPR2013-00085, and IPR2013-00086, involve the same parties and similar issues. The oral arguments for all six *inter partes* reviews were merged and conducted at the same time. A transcript of the oral hearing is included in the record as Paper 68, hereinafter "Transcript."

A. Related Proceeding

EMC indicates that the ’096 Patent is the subject of litigation titled *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.*, No. 6:11-cv-00660-LED (E.D. Tex.). Pet. 1.

B. The ’096 Patent

The ’096 Patent relates to a data processing system that identifies data items using substantially unique identifiers, otherwise referred to as True Names, which depend on all the data in the data item and only on the data in the data item. Ex. 1001, 1:44-48, 3:52-58, 6:20-24. According to the ’096 Patent, the identity of a data item depends only on the data and is independent of the data item’s name, origin, location, address, or other information not derivable directly from the data associated therewith. *Id*. at 3:52-58. The ’096 Patent also examines the identities of a plurality of data items in order to determine whether a particular data item is present in the data processing system. *Id*. at 3:59-62.

The ’096 Patent further discloses accessing data items by referencing their identities or True Names independent of their present location in the data processing system. *Id*. at 33:28-30. The actual data item or True file corresponding to a given data identifier or True Name is capable of residing anywhere on the data processing system, i.e., locally, remotely, offline, etc. *Id*. at 33:30-32. If a requested data item or True File is local with respect to the data processing system, a prospective user can access the data in the True File. *Id*. at 33:32-34. If a requested data item or True File is not local with respect to the data processing system, a prospective user may use the True File registry to determine the location of copies of the True File according to its given True Name. *Id.* at 33:34-38. However, if for some reason a prospective user cannot locate a copy of the requested data item or True File, the processor employed by the user may invoke the Request True File remote mechanism to submit a general request for the data item or True File to all the processors in the data processing system. *Id.* at 34:42-48.

C. Challenged Claim

Independent claim 1, along with dependent claims 2, 81, and 83, is challenged by EMC in this *inter partes* review and is reproduced below:

> 1. A computer-implemented method operable in a file system comprising a plurality of servers, the method comprising the steps of:
>
> (A) adding a data item to the file system, *the data item consisting of a sequence of non-overlapping parts, each part consisting of a corresponding sequence of bits*, by:
>
> (A1) for *each part* in said sequence of parts, *determining*, using hardware in combination with software, *a corresponding digital part identifier*, wherein each said digital part identifier for each said part is determined *based at least in part on a first function of all of the bits in the sequence of bits* comprising the corresponding part, the first function comprising a first hash function;
>
> (A2) *determining*, using a second function, *a digital identifier for the data item*, said digital data item identifier being *based, at least in part, on the contents of the data item*, wherein two identical data items in the file system will have the same digital data item identifier in the file

system, said second function comprising a second hash function;

(A3) storing each part in said sequence of parts on multiple servers of said plurality of servers in the file system;

(A4) storing first mapping data that maps the digital data item identifier of the data item to the digital part identifiers of the parts comprising the data item;

(A5) storing second mapping data that maps the digital part identifier of each part in said sequence of parts to corresponding location data that identifies which of the plurality of servers in the file system stores the corresponding part; and

(B) repeating step (A) for each of a plurality of data items; and

(C) attempting to access a particular data item in the file system by:

(C1) obtaining a particular digital data item identifier of the particular data item, said particular digital data item identifier of said particular data item being included in an attempt to access said particular data item in said file system;

(C2) attempting to match, using hardware in combination with software, said particular digital data item identifier of said particular data item with a digital data item identifier in said first mapping data; and

(C3) based at least in part on said attempting to match in step (C2), when said particular digital data item identifier obtained in step (C1) corresponds to an identifier in said first mapping data, using said first mapping data to determine a digital part identifier of each part comprising the particular data item;

(C4) using said second mapping data and at least one digital part identifier determined in step (C3) to determine

> location data that identifies which of the plurality of servers in the file system stores the corresponding at least one part of the particular data item;
>
> (C5) attempting to access at least one part of the particular data item at one or more servers identified in step (C4) as storing said at least one part.

Ex. 1001, 38:36-39:28 (emphasis added).

D. Prior Art Relied Upon

EMC relies upon the following prior art references:

Frederick W. Kantor, "*FWKCS (TM) Contents-Signature System Version 1.22*," FWKCS122.REF (Aug. 10, 1993) (Ex. 1004, hereinafter "Kantor").

Mahadev Satyanarayanan, "*Scalable, Secure, and Highly Available Distributed File Access*," 23 IEEE Computer 9-21 (May 1990) (Ex. 1005, hereinafter "Satyanarayanan").

E. Ground of Unpatentability

The Board instituted the instant trial based on the following ground of unpatentability:

Claims	Basis	References
1, 2, 81, and 83	§ 103(a)	Kantor and Satyanarayanan

II. ANALYSIS

A. Claim Construction

We begin our analysis by determining the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). An inventor may rebut that presumption by providing a definition of the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). In the absence of such a definition, limitations are not to be read from the specification into the claims. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993).

In the Decision on Institution, we construed the claim term "data item" to mean "sequence of bits," and observed that in the context of the specification, the meaning also includes one of the following: (1) the contents of a file; (2) a portion of a file; (3) a page in memory; (4) an object in an object-oriented program; (5) a digital message; (6) a digital scanned image; (7) a part of a video or audio signal; (8) a directory; (9) a record in a database; (10) a location in memory or on a physical device or the like; and (11) any other entity which can be represented by a sequence of bits. Dec. 10. The parties agree with that claim construction. Pet. 6-7; PO Resp.

1-2. As noted in the Decision on Institution, that claim construction is consistent with the specification. Dec. 9-10 (citing Ex. 1001, 1:56-57 ("the terms 'data' and 'data item' as used herein refer to sequences of bits."); *id*. at 1:56-61, 1:66–2:4). We discern no reason to deviate from that claim construction for the purposes of this decision.

B. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). In that regard, an obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418; *see also Translogic*, 504 F.3d at 1259.

We also recognize that prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *Paulsen*, 30 F.3d at 1480. Moreover, "it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." *In re Preda*, 401 F.2d 825, 826 (CCPA 1968). We analyze the instituted ground of unpatentability in accordance with the above-stated principles.

C. Claims 1, 2, 81, and 83 – Obviounesss over Kantor and Satyanarayanan

EMC asserts that claims 1, 2, 81, and 83 are unpatentable under 35 U.S.C. § 103(a) as obvious over Kantor and Satyanarayanan. Pet. 47-54. As support, EMC provides detailed explanations as to how each claim element, arranged as recited in the claim, is disclosed by Kantor, Satyanarayanan and/or the combination of both. *Id.* Additionally, EMC also directs our attention to the declaration of Dr. Clark. *Id.* (citing Ex. 1009).

PersonalWeb counters that Kantor fails to teach specific elements of claim 1 for which it is cited, that Kantor teaches away from the combination with Satyanarayanan, and that the proposed modification of Kantor would not have been obvious. PO Resp. 3-42. PersonalWeb also argues that claims 81 and 83 are not obvious over Kantor and Satyanarayanan, presenting separate arguments and relying on arguments made against the obviousness of claim 1. *Id.* at 42-50. PersonalWeb also alleges that Kantor is not a "printed publication" within the meaning of 35 U.S.C. § 102(b). *Id.*

at 51-56. In support of its argument, PersonalWeb proffers Mr. Todd Thompson's declaration (Ex. 2014).

Upon review of the parties' arguments and supporting evidence, we determine that EMC has demonstrated by a preponderance of the evidence that claims 1, 2, 81, and 83 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Kantor and Satyanarayanan. We also determine that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

Kantor

Kantor describes a method of identifying duplicate files. Ex. 1004, 2-4, 48-49. In particular, Kantor applies a hash function (*e.g.*, a cyclic residue check or cyclic redundancy check (CRC)) to each file within a zip file to obtain the contents signature for each file. *Id.* at 6-8, 48-49. Each contents signature is a string of bits generated from the contents of a file. *Id.*

For each zip file, Kantor creates zip-file contents signatures by hashing the contents signatures for the files contained within the zip file ("a hash of hashes"). *Id.* at 2, 9. As described by Kantor, this is done by "adding together all the 32_bit CRC's for the files in the zip file, modulo 2^32, separately adding together their uncompressed file_lengths modulo 2^32, and then arranging the two resulting hexadecimal numbers as a single structure." *Id.* at 9. Dr. Clark testifies that addition modulo 2^32 is another well-known simple hashing function that uses addition to calculate a value for a file based on the file's contents. Ex. 1009 ¶ 20. Kantor further compares the zip-file contents signatures to check for duplicate files. Ex. 1004, 2 of Preface, 5, 9.

According to Kantor, contents signatures and zip-file contents signatures are useful to identify files that have the same contents stored on the electronic bulletin board systems ("BBS"). Ex. 1004, 2 of Preface, 5, 9. For example, when uploading a zip file, the system determines whether that zip file already exists in the system using the zip-file contents signature, and then determines whether the inner files of that zip file already exist in the system using the contents signatures for the inner files. *Id*. at 9.

EMC has acknowledged that Kantor fails to disclose the underlying storage system of the BBS, and, thus, does not disclose that files are replicated on multiple servers, per claims 1, 2, 81, and 83. Pet. 52-53. Satyanarayanan discloses a network-based file replication system, where copies of files are stored at multiple servers (Ex. 1028, 447). EMC also argues that a person of ordinary skill would have found it obvious to modify Kantor to meet that limitation in view of Satyanarayanan. Pet. 53. On this record, we concur with the analysis of Dr. Clark, that it would have been obvious to combine Kantor and Satyanarayanan to provide more reliable storage systems for the BBS's files (Ex. 1009 ¶ 84).

Digital part identifiers for all parts of a data item/ based on all data

PersonalWeb argues that Kantor fails to disclose digital part identifiers for "each part" of a data item, or a data item identifier "based, at least in part, on the contents of the data item," because Kantor's "zipfile contents-signatures" are based on "contents-signatures" of the inner files of a zipfile, and are not based on all parts or all bits of the data item. PO Resp. 3-18. Because Kantor excludes some parts of the data item, i.e., the zip file, it is argued that it cannot teach the portions of claim 1 on which it is relied upon. *Id*. EMC counters that based on the claim construction for "data item," i.e., a "portion of a file," the inner files of the zip file are a portion of the file and are equivalent collectively to the "data item" of claim 1. Reply 1-2. We agree with EMC.

PersonalWeb assumes that the entire zip file in Kantor is equivalent to the "data item" in claim 1 (PO Resp. 3), but we are not persuaded that this is the sole, proper interpretation of Kantor. Claim 1 recites, in part, that "the data item consisting of a sequence of non-overlapping parts, each part consisting of a corresponding *sequence of bits*," and "for each part in said sequence of parts, determining . . . a corresponding digital part identifier . . . based at least in part on a first function of all of the bits in the *sequence of bits*." (emphasis added). We are persuaded that the inner files of the zip file in Kantor are equivalent to the claimed "data item" in that those files consist of a sequence of non-overlapping parts. In addition, each file of the inner files of the zip file consists of a sequence of bits, and a CRC of one of the individual files, in Kantor, is based on a function of "all of the bits in the sequence of bits" of the inner file.

Although PersonalWeb argues that the Petition found "that the 'data item' in Kantor is a ZIP file" (PO Resp. 6, citing Ex. 1009 ¶¶ 86, 100; Ex. 1029 ¶¶ 2, 8-10), the support for that statement, citing to Dr. Clark's declarations, also refers to "a sequence of non-overlapping parts" as the inner files within the zip file. Ex. 1009 ¶ 86, 100. We are persuaded that ground proffered in the Petition, and instituted in this proceeding, indicates that the "data item" of claim1 can be read as being equivalent to the inner files of the zip file in Kantor. Based on this, one need not use the *entire* zip file to meet the limitations of claim 1, if the inner files of the zip file constitute the data item. As a consequence, the fact that "a ZIP file includes much more than the individual 'files' therein," as argued by PersonalWeb (PO Resp. 9), is correct, but inapposite. Kantor's exclusion of other data or metadata in the zip file to determine the CRC is also not distinguishing. If the "data item" of claim 1 is taken as the inner files of the zip file in Kantor, then the fact that Kantor does not use every bit of the zip file does not distinguish it from claim 1.

This also comports with the overall purpose of the invention disclosed in the '096 Patent. The specification of the '096 Patent, in the "Summary of the Invention" section, provides that "the identity of the data item is independent of its name, origin, location, address, or other information not derivable directly from the data." Ex. 1001 3:56-58. Counsel for PersonalWeb argued at the oral hearing that "the things like File Name, et cetera, in the patents, the patent says -- makes clear they are not part of the data item. In the patent." Transcript 110. This is consistent with the view that that the inner files of a zip file can constitute a "data item," as claimed, in that such a construction would exclude metadata, i.e., the rest of the zip file, which can include name, location, etc. Limiting the "data item" to the inner files would meet the definition found in claim 1, with the inner files being "a portion of a file," and consist solely of a "sequence of non-overlapping parts." This is distinct from PersonalWeb's view (PO Resp. 3) that the data item be taken to be the whole zip file in Kantor.

PersonalWeb responds to this view of Kantor, although it alleges that "petitioner does not make this argument," and argues that "the express language of claim 1, and the Board's construction of 'data item,' preclude such an argument." PO Resp. 16. We do not agree. As discussed above, we find EMC has represented that the inner files of the zip file in Kantor are equivalent to the claimed "data item" in its Petition. PersonalWeb argues that "[b]ecause a data item must be a '*sequence* of bits,' one cannot pick and choose some bits of the ZIP file (the alleged 'data item'), *while excluding other intervening bits* of the ZIP file, to make up the alleged data item because the result would not be a "*sequence* of bits' as required by the claim." *Id*.

However, that interpretation depends on the meaning of "sequence of non-overlapping parts" whereby a sequence cannot have any intervening gaps. We are persuaded that the inner files of the zip file in Kantor, even with interstitial parts, such as local headers and directories, still form a sequence of non-overlapping parts. The inner files form a sequence of files that make up the zip file, with each file being a sequence of bits. The parts or files make up the sequence, even with metadata included between the parts. Considering the cited example from Dr. Clark's deposition (PO Resp. 17; Ex. 2016, 98), of a single file line of 100 people, ordinarily skilled artisans would consider that to be a sequence of people, and not need to look to any space left between the people as creating a non-sequence. One would not need to examine the intervening air, or mosquitoes, or dust that exists between the persons, because those elements would not be people. A sequence of persons need only look at the persons. Similarly, a sequence of inner files in Kantor can be a sequence, even if they have intervening "non-files" between them.

As such, we are persuaded that EMC has demonstrated that Kantor teaches that digital part identifiers are based on all parts of a data item, with the data item including a sequence of non-overlapping parts, each part consisting of a corresponding sequence of bits, and the digital part identifiers are based on all of the sequence of bits.

Kantor's 'y' procedure emphasizes the deficiencies of 'zcs'

PersonalWeb argues that the "y" procedure in Kantor (Ex. 1004, 55), wherein the CRC value is based on every byte in the zip file, illustrates that the "z" procedure, discussed above to compute zip-file contents signatures, does not apply a hash function to all of the data in the zip file. PO Resp. 18-21. PersonalWeb also argues that it would be fundamentally improper to switch between the "z" and "y" procedures to meet the limitations of claim 1 because the procedures are separate and distinct embodiments. *Id*. at 19-20. EMC counters that both disclosed procedures illustrate that whether to hash metadata is a mere design choice, and Kantor's preference to not hash metadata in the “z” procedure was made for the same reasons as made in the ’096 Patent to not hash metadata. Reply 5. We agree with EMC.

As discussed above, we are persuaded that EMC has demonstrated that the broadest reasonable construction of claim 1, consistent with the specification, does not demand the hashing of metadata, such that there is no need to combine different embodiments of Kantor to teach or suggest all of the elements of claim 1.

Kantor fails to teach or suggest sub-steps (A4)-(A5) of claim 1

PersonalWeb argues claim 1 requires “storing second mapping data that maps the digital part identifier of each part in said sequence of parts to corresponding location data.” PO Resp. 21. PersonalWeb argues that because a CRC is not computed for *all* parts of the zip file in Kantor, this element of claim 1 cannot be met. *Id*. We agree, however, with EMC that this argument is merely a restatement of PersonalWeb’s earlier argument, discussed above. We do not conclude that claim 1 requires the hashing of metadata contained in the zip file of Kantor, such that the sub-steps of claim 1, (A4) and (A5), can be met by Kantor. As such, we are persuaded that EMC has demonstrated that Kantor teaches or suggests the subject limitations of claim 1.

Applying a hash to each of the plurality of parts of the first data item

Claim 1 requires that "each said digital part identifier for each said part is determined based at least in part on a first function of all of the bits in the sequence of bits comprising the corresponding part, the first function comprising a first hash function." Based on this, PersonalWeb argues that claim 1 requires applying a first hash to parts of the data item to come up with the digital part identifiers, and that Kantor fails to disclose this and teaches away because Kantor applies the CRC hash to the uncompressed files before they are compressed and packaged into the zip file. PO Resp. 23. In other words, the CRC in Kantor is applied to different bit sequences (uncompressed files) than the bit sequences (compressed files) that make up the inner files of the zip file. *Id*. PersonalWeb also alleges that the bit sequence of an uncompressed file is much different that the bit sequence of a compressed version of that same file. *Id*. at 26. As such, PersonalWeb argues that Kantor fails to disclose an identifier based on a hash of the sequence of bits in the part of the data item, as called for in claim 1. *Id*. at 26.

EMC counters that nothing in the claims requires that the inner files of the zip file be compressed files. We are persuaded that, based on the present record, that a zip file can include uncompressed files and that Kantor can work with zip files regardless of the method or amount of compression. Ex. 1083, 263-265; Ex. 1089 ¶ 19-21; Ex. 1004, 9, 55. If the inner files of the zip file in Kantor are uncompressed, then the CRC hash values determined for the files before they become part of the zip file are the same as when the files are part of the zip file.

As such, we are persuaded that EMC has demonstrated that Kantor teaches or suggests the subject limitations of claim 1.

Determining the part identifiers in the file system

PersonalWeb argues that claim 1 requires the step of determining the part identifiers to be carried out "in a file system." PO Resp. 29-30. PersonalWeb argues that Kantor teaches away from claim 1 because the CRC values are determined outside the BBS. *Id.* PersonalWeb also alleges that Dr. Clark's testimony acknowledges that remote PCs are not part of the BBS when not logged into the BBS. *Id.*; Ex. 2016, 67, 101. EMC counters that ordinarily skilled artisans would appreciate that "a file system" could constitute the BBS, or it could constitute the BBS in combination with the computers communicating with the BBS, which is disclosed in Kantor. Reply 9. We agree with EMC.

The claim limitation "a file system" is not a limitation that has been construed specifically in this proceeding. Based on the deposition testimony, a "user's terminal or PC" would not be considered part of the BBS before login (Ex. 2016, 101), but that suggests that once it is connected, it would be considered a part of the BBS. Thus, PersonalWeb's analysis ignores the situation where a zip file is created by a user while connected to the BBS. We are persuaded that users connected to the BBS may form a new zip file, and thus also generate a zip-file contents signature, so that such a new file could be uploaded to the BBS. Dr. Clark also points out that the "Lookup" operation in Kantor demonstrates that the BBS and the users' computers can operate together as a file system. Ex. 1089 ¶ 27; Ex. 1004, 96. Therefore, we are not persuaded that Kantor fails to teach or suggest that the part identifiers are determined in the file system.

As such, we are persuaded that EMC has demonstrated that Kantor teaches or suggests the subject limitations of claim 1.

Kantor teaches away from adding zip files to multiple servers

Personalweb argues that claim 1 requires the storage of each part of the data item on multiple servers in the file system, which necessarily creates duplicate files in the system. PO Resp. 31. PersonalWeb argues that it would not have been obvious to have modified Kantor to accomplish this, even in view of Satyanarayanan, because Kantor teaches away by its very purpose of avoiding duplicate files in the system. *Id*. at 31-32. EMC counters that Kantor is concerned with avoiding *unwanted* duplicates, and Kantor is unconcerned with the mirroring of files on multiple servers because that is a function of the BBS. Reply 9-10. We agree with EMC.

Kantor generates and maintains a master list of the contents signatures called CSLIST.SRT, and the MULTIS feature is used to analyze the CSLIST, and identify and list the files for which multiple copies exist. Ex. 1004, 189. Thereafter, a word processor is used to add a "d" to the line of the MULTIS file for the files to be deleted. *Id*. Thus, if the user does not place the "d" on the line for the file, that duplicate file will remain on the system. Kantor does not require the elimination of all duplicates; it merely provides a mechanism that would allow for it. In view of the actual teachings of Satyanarayanan, namely that mirroring techniques can increase reliability and response times for requests for files (Ex. 1028, 450), we agree with EMC that it would have been obvious to mirror duplicate files that were not deleted in Kantor. Additionally, Dr. Dewar also agreed that mirroring technology was known. Ex. 1083, 114-115. Therefore, we are not persuaded that Kantor teaches away from mirroring of files on multiple servers.

Identifying files using contents signatures

Claim 1 recites "attempting to access a particular data item in the file system by: (C1) obtaining a particular digital data item identifier of the particular data item." Ex. 1001, 39:3-6. In its petition, EMC recognizes that the users typically request files based on the file names. Pet. 51. Nonetheless, EMC asserts that a person having ordinary skill in the art would have found it obvious to modify the electronic Bulletin Board Systems commands, including the download and read commands, to identify files using contents signatures or zip-file contents signatures, instead of file names. *Id*. at 51-52 (citing Ex. 1009 ¶ 83). According to EMC, "this would facilitate integrity checking by more precisely specifying the file of interest by its content, and thus improve accuracy." *Id*. at 51. Dr. Clark testifies that such a modification would provide a more efficient and context-free means for accessing and sharing files. Ex. 1009 ¶ 83.

PersonalWeb counters that it would not have been obvious to modify Kantor so that the read and download requests would accept contents signatures to identify files. PO Resp. 35-42. PersonalWeb alleges that Kantor fails to teach or suggest the alleged modification, and fails to provide any suggestion or motivation for the alleged modification. *Id.* at 35-39 (citing Ex. 2017 ¶¶ 62-63). PersonalWeb further submits that Kantor does not disclose any problems with the use of conventional file names for the read and download requests. *Id.* at 39-40. Additionally, PersonalWeb argues that Kantor teaches away from replacing conventional file names with contents signatures for identifying files, because "Kantor intentionally designed his contents-signatures so that certain different files would have the same signature." *Id.* at 40-42 (citing Ex. 1004, 3, 51; Ex. 2017 ¶¶ 64-66).

We are not persuaded by PersonalWeb's arguments. As to PersonalWeb's arguments that Kantor does not provide a motivation for the modification (*id.* at 36), a rationale to combine the prior art teachings does not have to be found explicitly in the prior art, itself. *See In re Kahn*, 441 F.3d 977, 987 (Fed. Cir. 2006) (A "motivation to combine the relevant prior art teachings does not have to be found explicitly in the prior art."). We also are not persuaded by PersonalWeb's argument that there would have not been a logical reason to modify Kantor in the manner alleged by EMC, other than impermissible hindsight (PO Resp. 36). As discussed above, EMC asserts that it would have been obvious to modify the read and download commands to identify files using contents signatures instead of file names. Pet. 51-52 (citing Ex. 1009 ¶ 83). EMC takes the position that "this would facilitate integrity checking by more precisely specifying the file of interest by its content, and thus improve accuracy." *Id.* Dr. Clark testifies that such a modification would provide a more efficient and context-free means for accessing and sharing files. Ex. 1009 ¶ 83. EMC's position and Dr. Clark's testimony are consistent with Kantor's disclosure that using contents signatures, instead of file names, to find and delete duplicate files would increase system efficiency by reducing storage cost and system time for locating and managing files. Ex. 1004, Preface, 5, 9, 205-206. As such, we conclude that EMC has articulated a sufficient reason to combine the teachings of Kantor.

Also, we are not persuaded by PersonalWeb's argument that the proposed modification is not enabled and its argument that EMC fails to explain how the proposed modification could have been carried out to yield a predictable result. PO Resp. 36-39. EMC specifically explains that Kantor's Precheck and Lookup operations provide examples of user commands that utilize contents signatures. Pet. 46 (citing Ex. 1004, 97, 173; Ex. 1009 ¶ 83). For instance, Kantor describes the Precheck operation as a software utility running on the electronic Bulletin Board Systems for identifying files that already uploaded in the system by using their contents signatures. Pet. 51-52 (citing Ex. 1004, 173). Dr. Clark explains that Kantor's Lookup operation permits users to submit a request containing a contents signature to determine where the corresponding file is located on the system. Ex. 1009 ¶ 83 (citing Ex. 1004, 96-97). Dr. Clark further testifies the system as modified would have utilized one of those contents signatures for the inner files in a download request to obtain the particular inner file that is associated with the contents signature. *Id*. Upon review of the parties' contentions and supporting evidence, we agree with EMC that Dr. Clark merely relies on the disclosure of Kantor (Ex. 1004, 96-97), and not LOOKUP.DOC and PRECHECK.DOC files as alleged by PersonalWeb. For the foregoing reasons, we determine that EMC has explained sufficiently how the proposed modification could have been carried out to yield a predictable result.

PersonalWeb's argument that Kantor does not teach or suggest the alleged modification is unpersuasive, because an obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418. PersonalWeb's argument overlooks "the fundamental proposition that obvious variants of prior art references are themselves part of the public domain." *Translogic,* 504 F.3d at 1259. Moreover, we observe that the asserted ground of unpatentability is based on the *combination* of Kantor's teaching of using contents signatures to identify files with Kantor's teaching of requesting files. It is well settled that nonobviousness cannot be established by attacking each prior art teaching individually where, as here, the ground of unpatentability is based upon a combination of different teachings in the prior art. *See In re Keller*, 642 F.2d 413, 426 (CCPA 1981). Rather, the test for obviousness is whether the combination of prior art teachings, taken as a whole, would have suggested the patentees' invention to a person having ordinary skill in the art. *See In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

In light of Kantor, a person of ordinary skill in the art would have recognized how to calculate contents signatures and zip-file contents signatures and how to use them to identify files. *See, e.g.*, Ex. 1004, Preface, 5-9. A person with ordinary skill in the art also would have appreciated the benefit of using contents signature and zip-file contents signatures that are generated based on the contents of the files, rather than *file names*, for identifying files accurately. *Id*. The mere substitution of contents signatures and zip-file contents signatures for *file names* in read and download requests predictably uses prior art elements according to their established functions. Such a substitution is an obvious improvement. *See KSR*, 550 U.S. at 417 (The simple substitution of one known element for another is likely to be obvious if it does no more than yield predictable results.). Moreover, PersonalWeb has not provided sufficient evidence that such a substitution is beyond the level of a person with ordinary skill in the art. *See Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007).

PersonalWeb's teaching away argument is misplaced, as it fails to recognize that the cited portion of Kantor specifically explains that the different files that allegedly have the same signature files also have the *same contents*. *See* Ex. 1004, 3 ("[T]he same file contents . . . will have the same zipfile contents signature."). In fact, that is one of the reasons why using contents signatures or zip-file contents signature, instead of file names, to identify files is more accurate. Ex. 1004, Preface, 5, 9. Notably, files that have the *same contents* would be identified as duplicates, and files that have *different contents* would be identified as different files, regardless of whether they have different file names. *Id*. As Kantor notes, finding and deleting duplicate files would improve system efficiency. *Id*.

Obviousness of Claims 2, 81, and 83

PersonalWeb discusses the subject matter of claim 2 only briefly, arguing that in addition to Kantor not teaching elements of claim 1, Kantor also fails to teach elements of claim 2, namely a second hash function applied to respective digital part identifiers in determining the digital identifier. PO Resp. 19, 29. We do not find those brief arguments to be persuasive and conclude that EMC has demonstrated that claim 2 is obvious over Kantor and Satyanarayanan. Pet. 56. PersonalWeb also separately argues the subject matters of claims 81 and 83 (PO Resp. 42-46), but we concur with EMC (Reply 13) that those arguments rely on the same claim limitations and make the same arguments already discussed above with respect to claim 1. We do not find such arguments any more persuasive with respect to claims 81 and 83.

Evidence of non-obviousness

PersonalWeb further submits that its evidence of non-obviousness rebuts EMC's evidence of obviousness. PO Resp. 50-51. In support of its argument, PersonalWeb directs our attention to three licensing agreements, as well as the declaration of Mr. Kevin Bermeister. *Id*. at 12 (citing Exs. 2010-12; Ex. 2009 ¶¶ 3-9). PersonalWeb argues that each license granted to a third party was not for the purpose of settling a patent infringement suit. *Id*.

In its Reply, EMC contends that PersonalWeb has failed to establish a sufficient nexus between claims 1, 2, 81, and 83 of the ’096 Patent and the above-identified license agreements. Reply 13-14. EMC argues that each of the licenses granted rights to more than just claims 1, 2, 81, and 83, and involved related parties with interlocking ownership and business interests. *Id*. We agree with EMC that PersonalWeb has failed to establish the requisite nexus between the licensing agreements and claims 1, 2, 81, and 83.

A party relying on licensing activities as evidence of non-obviousness must demonstrate a nexus between those activities and the subject matter of the claims at issue. *In re GPAC Inc.*, 57 F.3d 1573, 1580 (Fed. Cir. 1995). Further, without a showing of nexus, “the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness” when there is a strong ground of unpatentability based on obviousness. *SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1358 (Fed. Cir. 2000); *see Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004).

The evidence of non-obviousness presented by PersonalWeb falls short of demonstrating the required nexus. Neither PersonalWeb nor the declaration of Mr. Bermeister (Ex. 2009) establishes that the licensing agreements (Exs. 2010-12) are directed to the claimed subject matter recited in claims 1, 2, 81, and 83. For instance, PersonalWeb does not present credible or sufficient evidence that the three licensing agreements arose out of recognition and acceptance of the claimed subject matter recited in claims 1, 2, 81, and 83. In the absence of an established nexus with the claimed invention, secondary consideration factors are entitled little weight, and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985). Furthermore, even if we assume that above-identified licenses establish some degree of industry respect for the claimed subject matter recited in claims 1, 2, 81, and 83, that success is outweighed by the strong evidence of obviousness over Kantor and Satyanarayanan discussed above.

Based on the record before us, including the evidence of obviousness presented by EMC and the evidence of secondary considerations regarding licensing activities presented by PersonalWeb, we conclude that EMC has demonstrated by a preponderance of the evidence that claims 1, 2, 81, and 83 would have been obvious over the combination of Kantor and Satyanarayanan.

Whether Kantor is a "printed publication"

In its petition, EMC takes the position that Kantor is a "printed publication" under 35 U.S.C. § 102(b). Pet. 5. EMC asserts that Kantor has been publicly available since August 1993, which is prior to the critical date, April 11, 1995, one year before the earliest priority date claimed by the '096

Patent. *Id*. To substantiate its position, EMC explains that Kantor is "a published manual that describes a software program called the Frederick W. Kantor Contents-Signature System Version 1.22 ('FWKCS')." *Id*. at 47 (citing Ex. 1004, Title Page). EMC maintains that Dr. Frederick Kantor distributed Kantor—the user manual (version 1.22), the version relied upon by EMC (*see* Ex. 1004)—with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where Kantor could be downloaded by anyone. Pet. 5, n. 3 (citing Ex. 1004, 3, 158-59). According to EMC, Kantor was accessible to others in the relevant community of the users and system operators of electronic Bulletin Board Systems. *Id*. In support of its position, EMC proffers a declaration of Mr. Michael A. Sussell (Ex. 1050) and declarations of Mr. Jason S. Sadofsky (Ex. 1078; Ex. 1088).

In its patent owner response, PersonalWeb counters that Kantor is not a "printed publication." PO Resp. 51-56. In particular, PersonalWeb alleges that EMC has not established that the specific version of Kantor existed prior to the critical date. *Id*. at 52. PersonalWeb contends that there is no evidence that Kantor was disseminated publicly, catalogued, or indexed in a meaningful way. *Id*. at 52-53. It is PersonalWeb's view that EMC fails to establish that one with ordinary skill in the art, exercising reasonable diligence, would have located Kantor prior to the critical date. *Id*. at 51.

We have reviewed the parties' arguments and supporting evidence. Based on the evidence before us, we are not persuaded by PersonalWeb's arguments. Rather, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b).

The determination of whether a given reference qualifies as a prior art "printed publication" involves a case-by-case inquiry into the facts and circumstances surrounding the reference's disclosure to members of the public. *In re Klopfenstein*, 380 F.3d 1345, 1350 (Fed. Cir. 2004). The key inquiry is whether the reference was made "sufficiently accessible to the public interested in the art" before the critical date. *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989); *In re Wyer*, 655 F.2d 221, 226 (CCPA 1981). "A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it." *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006).

Indexing is not "a necessary condition for a reference to be publicly accessible," but it is only one among many factors that may bear on public accessibility. *In re Lister*, 583 F.3d 1307, 1312 (Fed. Cir. 2009). In that regard, "while often relevant to public accessibility, evidence of indexing is not an absolute prerequisite to establishing online references . . . as printed publications within the prior art." *Voter Verified, Inc. v. Premier Election Solutions, Inc.*, 698 F.3d 1374, 1380 (Fed. Cir. 2012).

Contrary to PersonalWeb's assertion that Kantor did not exist prior to the critical date and there is no evidence that Kantor was disseminated publicly, Kantor itself shows a copyright date of "1988-1993" and a posted date of "1993 August 10." Ex. 1004, Title Page, the first page after the Title Page ("All of the programs and documents, comprising the entire contents of this Authenticity Verification Zip file FWKCS122.ZIP, together with this Zipfile itself, are, in accordance with their respective dates of creation or revision, (C) Copyright Frederick W. Kantor 1988-1993."). Kantor also states:

> The FWKCS(TM) Contents_Signature System has become a robust platform for supporting contents_signature functions. FWKCS provides many functions and options for application in a public, commercial, school, institutional, or governmental environment. Extensive technical support is of special value in helping such users to benefit more fully from these many features.
>
> Registered FWKCS hobby BBS users are able to receive a modest amount of assistance, and are invited to participate in the FWKCS conference on The Invention Factory BBS, echoed via Execnet.
>
> Commercial, school, institutional, and governmental users, with their special support needs, are invited to discuss terms for obtaining such assistance.
>
> . . . .
>
> To get a new version of FWKCS, download FWKCSnnn.ZIP from The Invention Factory BBS, where nnn is the new version number without a decimal point. These special downloads are available at no fee, from a 43_line hunt_up group of USR Dual Standard modems, at 2400-16800 bits/sec (including V32.bis).

Ex. 1004, 158-159. It is clear from Kantor that, during the 1988-1993 timeframe, Dr. Kantor had posted many versions of his software and user manual—including Kantor (version 1.22),, the version relied upon by EMC (Ex. 1004)—on electronic Bulletin Board Systems.

Mr. Sussell, the co-owner and system operator of the Invention Factory Bulletin Board System, testifies that the Invention Factory Bulletin Board System is a computer system that allows users to share files, messages, and articles, as well as search, upload, and download files. Ex. 1050 ¶¶ 3-4. According to Mr. Sussell, he and his wife launched the Invention Factory Bulletin Board System in 1983, and it had over 3,000 subscribers by mid-1993. *Id.* ¶ 6. Mr. Sussell testifies that, by 1993, the system provided all users keyword search functionality and access to various descriptive and meaningful directories. *Id.* ¶¶ 8-10.

More importantly, Mr. Sussell testifies that the Invention Factory Bulletin Board System "extensively utilized and hosted current versions of FWKCS software on its [Bulletin Board System]," and "made publicly accessible and available the complete FWKSC ZIP file that contained both the software as well as related documentation such as user manuals" prior to the critical date. *Id.* ¶ 15; *see id.* ¶¶ 16-27. Specifically, Mr. Sussell testifies that users would have found Kantor by performing keyword searches on the Invention Factory Bulletin Board System. *Id.* ¶ 21. Mr. Sussell also indicates that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about Dr. Kantor's software on the "Welcome" screen, and made the FWKCS Zip file available in four different directories. *Id.* ¶¶ 18-20. Mr. Sussell further testifies that computer disks that contain the FWKCS Zip file were distributed at various Bulletin Board System conferences. *Id.* ¶ 18.

Mr. Sadofsky, a technology archivist and software historian, testifies that he personally verified the authenticity of Kantor—the user manual (version 1.22), the version relied upon by EMC (Ex. 1004)—by comparing it with a "1993 archived" version, and determined that Kantor is identical to the "1993 archived" version. Ex. 1078 ¶¶ 14-17. Mr. Sadofsky testifies that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id.* ¶ 16; Ex. 1088 ¶¶ 10-11; Ex. 2014 ¶ 5. According to Mr. Sadofsky, Kantor was publicly accessible prior to the critical date. *Id.*

PersonalWeb also asserts that Kantor was buried and hidden in the zip file in a manner such that "it would not have been located and accessed by persons interested and ordinarily skilled in the art exercising reasonable diligence even if they had access to the ZIP file." PO Resp. at 53-54 (citing Ex. 2014). However, PersonalWeb's supporting evidence, Mr. Thompson's declaration (Ex. 2014), does not substantiate PersonalWeb's assertion. Upon review of Mr. Thompson's declaration, we observe that Mr. Thompson downloaded the FWKCS Zip file—the zip file that contains the software and Kantor, the user manual—without any difficultly. Ex. 2014 ¶ 5. Significantly, Mr. Thompson did not follow the instructions provided with the zip file, nor did he use the appropriate computer environment (DOS 3.0 or an IBM OS/2 2.0) that was used normally in 1993-1994 timeframe, but instead he used non-compatible software (DOS 8.0 and 32-bit Windows XP operating system that was released in 2001). Ex. 2014 ¶¶ 6-11; Ex. 1088

¶¶ 5, 14. Once he followed the instructions and unzipped the FWKCS Zip file, Mr. Thompson located Kantor without difficulty. Ex. 2014 ¶¶ 20-22.

Mr. Sadofsky confirms that the README.TXT file provides simple instructions and, if a user follows the instructions and uses the operating system that was used normally in 1993-1994 timeframe, the user could locate Kantor without difficulty. Ex. 1088 ¶¶ 13-17. In fact, Mr. Sadofsky demonstrated, in his declaration, several relatively easy ways for a user to access Kantor—with or without installing the software, and with or without help screens. Ex. 1088 ¶¶ 8-16 (II. README.TXT); ¶¶ 17-20 (III. GETLOOK.BAT); ¶¶ 21-22 (IV. FWKCS122 Start Screen and In-Program Help). Based on the evidence before us, we determine that Kantor was available to the extent that persons interested and ordinarily skilled in the art, exercising reasonable diligence, could locate it.

PersonalWeb's argument that EMC's witnesses personally did not post or review Kantor prior to the critical date also is unavailing. PO Resp. 52-54 (citing Ex. 2015, 52-55; Ex. 2013, 29-30; Ex. 2016, 98). It is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986) (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible"); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'"); *In re Bayer*, 568 F.2d 1357, 1361 (CCPA 1978) (A reference constitutes a "printed publication" under 35 U.S.C. § 102(b) as long as a presumption is raised that the portion of the public concerned with the art would know of the invention.).

The evidence on this record clearly support that Kantor was posted on a publicly accessible site—the Invention Factory Bulletin Board System—well known to those interested in the art, and could be downloaded and retrieved from that site, and, therefore, Kantor, an electronic publication, is considered a "printed publication" within the meaning of 35 U.S.C. § 102(b). *See Wyer*, 655 F.2d at 226 (An electronic publication, including an on-line database or Internet publication, is considered to be a "printed publication" "upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it and recognize and comprehend therefrom the essentials of the claimed invention without need of further research or experimentation.").

For the foregoing reasons, we determine that EMC has demonstrated by a preponderance of the evidence that Kantor is a "printed publication" within the meaning of 35 U.S.C. § 102(b). Therefore, EMC may rely upon Kantor for its asserted ground of unpatentability under 35 U.S.C. § 103(a).

D. EMC's Motion to Exclude

EMC seeks to exclude the following exhibits: (1) three license agreements (Exs. 2010-12); (2) Mr. Bermeister's declarations (Exs. 2009, 2018) relating to those license agreements; and (3) Mr. Thompson's declaration (Ex. 2014). Paper 54 ("Pet. Mot."). PersonalWeb filed the license agreements and Mr. Bermeister's declarations as evidence of non-obviousness to rebut EMC's assertion that claims 1, 2, 81, and 83 would have been obvious over the combination of Kantor and Satyanarayanan. PO Resp. 50-51. As to Mr. Thompson's declaration, PersonalWeb proffered that evidence to support its assertion that Kantor—a user manual that was disseminated publicly with the software in a zip file—was not made sufficiently accessible to a person interested and ordinarily skilled in the art. *Id*. at 54-55. PersonalWeb opposes EMC's motion to exclude. Paper 57. In response, EMC filed a reply to PersonalWeb's opposition to its motion to exclude. Paper 62.

With respect to the license agreements and Mr. Bermeister's declarations (Exs. 2009-2012, 2018), EMC argues that they are irrelevant under Federal Rule of Evidence 402[2], highly prejudicial, confusing, and misleading under Federal Rule of Evidence 403. *Id*. at 8-13. As to Mr. Thompson's declaration, EMC argues that it should be excluded under Federal Rule of Evidence 402. *Id*. at 14-15. Specifically, EMC alleges that:

[2] As stated in 37 C.F.R. § 42.62, the Federal Rules of Evidence generally apply to proceedings, including *inter partes* reviews.

(1) Mr. Thompson does not possess the skill of a person of ordinary skill in the art (*id*. at 14-15 (citing Ex. 1086, 13-14)); (2) Mr. Thompson did not use compatible software from the relevant time period (*id*. at 15 (citing Ex. 1086, 40-41; Ex. 2014, 4, 6)); and (3) Mr. Thompson did not follow the instructions provided with the zip file (*id*. at 15 (citing Ex. 1086, 32-35)).

The current situation does not require us to assess the merits of EMC's motion to exclude. As discussed above, even without excluding PersonalWeb's supporting evidence, we have determined that Kantor is a "printed publication" under 35 U.S.C. § 102(b), and EMC has demonstrated, by a preponderance of the evidence, that claims 1, 2, 81, and 83 are unpatentable over the combination of Kantor and Satyanarayanan.

Accordingly, EMC's motion to exclude evidence is *dismissed* as moot.

E. PersonalWeb's Motion to Exclude

PersonalWeb seeks to exclude the following items of evidence: (1) Kantor (Ex. 1004); (2) certain documents (Exs. 1047-1049, 1052-1055, 1074, 1075, 1080-1082) and the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) regarding those documents; (3) the declarations of Messrs. Sussell and Sadofsky regarding Kantor (Exs. 1050, 1078, 1088) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66); and (4) Clark's rebuttal declaration (Ex. 1089 ¶¶ 26-27, 30). Paper 53 ("PO Mot.").

EMC opposes PersonalWeb's motion to exclude. Paper 60 ("Opp."). In response, PersonalWeb filed a reply to EMC's opposition to its motion to exclude. Paper 63 ("PO Reply"). For the reasons stated below, PersonalWeb's motion to exclude is *denied*.

Kantor

PersonalWeb alleges that Kantor should be excluded as unauthenticated and inadmissible hearsay under Federal Rules of Evidence 901 and 902. PO Mot. 1, 6. In particular, PersonalWeb argues that "[n]o witness of record has personal knowledge of Kantor existing prior to [the critical date], and electronic data such as Kantor is inherently untrustworthy because it can be manipulated from virtually any location at any time." *Id*. at 2-4. According to PersonalWeb, the dates provided by Kantor are inadmissible hearsay because Kantor is not self-authenticating. *Id*. at 2, 5-6.

EMC argues that Kantor has been authenticated under Federal Rules of Evidence 901, and that the document is not hearsay, because it is being offered for what it describes—not for the truth of its disclosures. Opp. 1-10. In particular, EMC disagrees with PersonalWeb that Kantor cannot be authenticated without direct testimony from a witness with personal knowledge that Kantor existed prior to the critical date. Opp. 1. EMC asserts that it need "only produce evidence 'sufficient to support a finding' that the reference 'is what the proponent claims it is.'" *Id*. at 1-2 (citing Fed. R. Evid. 901(a)). EMC also contends that testimony from Messrs. Sussell and Sadofsky provides sufficient evidence to authenticate Kantor. Opp. 1-5 (citing Exs. 1050, 1078, 1088).

In its reply, PersonalWeb argues that Federal Rules of Evidence identified by EMC are not applicable to Kantor, because Mr. Sussell did not post or review Kantor prior to critical date. PO Reply 1-5 (citing Ex. 2015, 32-36, 55, 55, 65). PersonalWeb also alleges that Kantor's authenticity is suspicious, as electronic data are inherently untrustworthy and there is no chain of custody. *Id*.

We have considered PersonalWeb's arguments as well as EMC's contentions and supporting evidence. We are not persuaded that Kantor should be excluded.

At the outset, we disagree with PersonalWeb's position that a witness cannot authenticate a document, unless the witness is the author of the document or the witness has reviewed the document prior to the critical date. Federal Rule of Evidence 901(a) states that the authentication requirement is satisfied if the proponent presents "evidence sufficient to support a finding that the item is what the proponent claims it is." Therefore, neither a declaration from the author, nor evidence of someone actually viewing the document *prior to critical date*, is required to support a finding that the document is what it claims to be. *See also Hall*, 781 F.2d at 899 (concluding "that competent evidence of the general library practice may be relied upon to establish an approximate time when a thesis became accessible."); *Wyer*, 655 F.2d at 226 (Notwithstanding that there is no evidence concerning actual viewing or dissemination of any copy of the Australian application, the court held that "the contents of the application were sufficiently accessible to the public and to persons skilled in the pertinent art to qualify as a 'printed publication.'").

Further, it is well settled that an uninterrupted chain of custody is not a prerequisite to admissibility, but rather gaps in the chain go to weight of the evidence. *U.S. v. Wheeler*, 800 F.2d 100, 106 (7th Cir. 1986); *see also U.S. v. Aviles*, 623 F.2d 1192, 1198 (7th Cir. 1980) ("If the trial judge is satisfied that in reasonable probability the evidence has not been altered in any material respect, he may permit its introduction.") (Citation omitted). There is a strong public policy for making all information filed in a quasi-judicial administrative proceeding available to the public, especially in an *inter partes* review, which determines the patentability of a claim in an issued patent. It is within the Board's discretion to assign the appropriate weight to be accorded to evidence.

Although Messrs. Sussell and Sadofsky personally did not post or review the particular version of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—prior to the critical date, they have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. *See Hall*, 781 F.2d at 899; *Wyer*, 655 F.2d at 226; *Bayer*, 568 F.2d at 1361.

Notably, Mr. Sussell, the co-founder and system operator of the Invention Factory Bulletin Board System, testifies that Dr. Kantor released the first version of his software on the Invention Factory Bulletin Board System in the 1980s, and the system continuously utilized and hosted current versions of the software and user manuals. Ex. 1050 ¶¶ 3, 13, 15. Mr.

Sussell also testifies that the Invention Factory Bulletin Board System advertised Dr. Kantor's software to its users by including information about Dr. Kantor's software on the "Welcome" screen, and made FWKCS Zip file—a zip file that contains both the software and user manual—publicly accessible and available under four different directories. *Id.* ¶ 18. According to Mr. Sussell, the Invention Factory Bulletin Board System had over 3,000 subscribers, in the 1993 timeframe, and all of the users had the capability to perform keyword searches to retrieve FWKCS Zip file. *Id.* ¶¶ 6, 21.

Although we are cognizant that electronic documents downloaded from websites normally are not self-authenticating, it has been recognized that "[t]o authenticate printouts from a website, the party proffering the evidence must produce some statement or affidavit from someone with knowledge of the website . . . for example a web master or someone else with personal knowledge would be sufficient." *St. Luke's Cataract and Laser Institute v. Sanderson*, 2006 WL 1320242, *2 (M.D. Fla. 2006) (citing *In re Homestore.com, Inc. Sec.Litig.*, 347 F. Supp. 2d 769, 782 (C.D. Cal. 2004)) (quotation marks omitted); Ex. 2024; *see also Market-Alerts Pty. Ltd. v. Bloomberg Finance L.P.*, 922 F. Supp. 2d 486, 493, n.12 (D. Del. 2013) (citing *Keystone Retaining Wall Sys., Inc. v. Basalite Concrete Prods., LLC,* 2011 WL 6436210, at *9 n.9 (D. Minn. 2011)) (documents generated by a website called the Wayback Machine have been accepted generally as evidence of prior art in the patent context); *U.S. v. Bansal*, 663 F.3d 634, 667-68 (3d. Cir. 2011) (concluding that the screenshot images from the

Internet Archive were authenticated sufficiently under Federal Rule of Evidence 901(b)(1) by a witness with personal knowledge of its contents, verifying that the screenshot the party seeks to admit are true and accurate copies of Internet Archive's records).

Here, Mr. Sadofsky, who is a technology archivist and software historian and currently is an archivist for the Internet Archive, testifies that he launched the website textfiles.com and a subdomain cd.textfiles.com to collect software, data files, and related materials from Bulletin Board Systems. Ex. 1078 ¶¶ 9-11. According to Mr. Sadofsky, textfiles.com and cd.textfiles.com are dedicated to preserving, archiving, and providing free access to unaltered historical software programs and information that initially were made available on the Bulletin Board System. *Id.* Mr. Sadofsky states that he previously archived the FWKCS Zip file (FWKCS122.ZIP) that contains Dr. Kantor's software and user manual to cd.textfiles.com from his own copy of the *Simtel MSDOS Archive,* October 1993 Edition, Walnut Creek CD-ROM. *Id.* ¶ 14 (citing Ex. 1049).
Mr. Sadofsky also testifies that he personally verified the authenticity of Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—by comparing it with the "1993 archived" version and determined that Kantor is identical to the "1993 archived" version. Ex. 1078 ¶¶ 13-15. Mr. Sadofsky confirms that the source file of the "1993 archived" version has a timestamp of August 10, 1993, at 1:22 AM. *Id.* ¶ 16; Ex. 1088 ¶¶ 10-11; Ex. 2014 ¶ 5. Mr. Sadofsky concludes that Kantor was publicly accessible prior to the critical date. Ex. 1078 ¶¶ 13, 16.

Moreover, we agree with EMC that Kantor also has been authenticated as an "ancient document" under Federal Rule of Evidence 901(b)(8).[3] Opp. 6-7 Kantor is "at least 20 years old and can be found in . . . an October 1993 *Simtel* CD-ROM – a place where an authentic 20-year old document distributed through a [Bulletin Board System] would likely be." *Id.*; Ex. 1072 ¶¶ 7-8; *see also* Fed. R. Evid. 901(b)(8) 2012 Adv. Comm. Note ("The familiar ancient document rule of the common law is extended to include data stored electronically or by other similar means."). Moreover, testimony of Messrs. Sussell and Sadofsky has established sufficiently that Kantor is in a condition that creates no suspicion about its authenticity. Exs. 1050, 1078, 1088.

PersonalWeb does not present sufficient or credible evidence to the contrary. Based on the evidence before us, we determine that Kantor has been authenticated under Federal Rules of Evidence 901(b)(1), (b)(3), (b)(4), and (b)(8) to warrant its admissibility.

PersonalWeb's hearsay argument regarding Kantor also is unavailing. As EMC notes (Opp. 7), "[p]rior art references are not hearsay because they are offered for what they *describe*, and *not* to prove the truth of the matters asserted." *See, e.g., Joy Techs., Inc. v. Manbeck*, 751 F. Supp. 225, 233 n.2

[3] Fed. R. Evid. 901(b)(8). Evidence About Ancient Documents or Data Compilations. For a document or data compilation, evidence that it:
(A) is in a condition that creates no suspicion about its authenticity;
(B) was in a place where, if authentic, it would likely be; and
(C) is at least 20 years old when offered.

(D.D.C. 1990), *judgment aff'd*, 959 F.2d 226 (Fed. Cir. 1992); Fed. R. Evid. 801(c) 1997 Adv. Comm. Note ("If the significance of an offered statement lies solely in the fact that it was made, no issue is raised as to the truth of anything asserted, and the statement is not hearsay."). Therefore, Kantor is not hearsay under Federal Rule of Evidence 801(c).

We further agree with EMC that the posted date of "1993 August 10" or the copyright date of "1988-1993" on the Title page of Kantor is not a basis for excluding Kantor, as testimony from Messrs. Sussell and Sadofsky sufficiently establishes that Kantor existed as of August 10, 1993, prior to the critical date. Opp. 9-11. More importantly, the computer-generated timestamp—August 10, 1993, at 1:22 AM—of the "1993 archived" version of Kantor (Ex. 1078 ¶¶ 14-15; Ex. 1088 ¶¶ 10-11; Ex. 2014 ¶ 5) also independently corroborates Kantor's existence as of August 10, 1993. *See, e.g., U.S. v. Khorozian*, 333 F.3d 498, 506 (Fed. Cir. 2003) (concluding that an automatically generated time stamp on a fax was not a hearsay statement because it was not uttered by a person). Accordingly we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Kantor as impermissible hearsay.

For the foregoing reasons, we decline to exclude Kantor.

Documents Corroborating Witnesses' Knowledge and Recollections

PersonalWeb asserts that certain documents submitted by EMC (Exs. 1047-1049, 1052-1055, 1074, 1075, 1080-1082) and the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) regarding those documents should be excluded because the documents have not been authenticated properly and are inadmissible hearsay. PO Mot. 6-10. PersonalWeb argues that EMC "has not established that any of these documents existed prior to the critical date, and no witness has personal knowledge of their alleged existence prior to April 11, 1995." *Id*. at 7. PersonalWeb further maintains that the documents that are Exhibits 1053, 1054, 1074, and 1075 are irrelevant, prejudicial, and confusing, as they discuss a version of Kantor different than the version relied upon by EMC (version 1.22, Ex. 1004). *Id*. at 8.

EMC responds that its witnesses provided those "documents concerning Kantor to corroborate their independent knowledge and recollections." Opp. 9. EMC asserts that the documents have been authenticated under Federal Rules of Evidence 901-902 and fall within a hearsay exception under Federal Rules of Evidence 803-807. *Id*. at 10-11. We are persuaded by EMC's arguments.

As the movant, PersonalWeb has the burden of proof to establish that it is entitled to the requested relief. 37 C.F.R. § 42.20(c). As discussed previously, we disagree with PersonalWeb that documents cannot be authenticated without direct testimony from the author or a witness who actually reviewed the documents prior to the critical date. *See* Fed. R. Evid. 901(a). Significantly, PersonalWeb's motion does not contain any sufficient explanation why each document should be excluded. For instance, PersonalWeb does not explain adequately why the declaration of Mr. Sussell (Ex. 1050 ¶¶ 6, 8, 18, 27) is not sufficient to authenticate Exhibits

1052-1055, 1074, and 1075, or why the declarations of Mr. Sadofsky (Ex.1078 ¶¶ 7-17; Ex. 1088 ¶¶ 10-16) are not sufficient to authenticate Exhibits 1047-49 and 1080-1082. *See* Fed. R. Evid. 901(b)(1).[4] Nor does PersonalWeb explain sufficiently why the following documents are not self-authenticated: (1) Exhibits 1047-1049 and 1052 that include articles containing LexisNexis® trade inscriptions; (2) Exhibits 1074 and 1075 that include Usenet newsgroup periodicals containing Usenet trade inscriptions; and (3) Exhibit 1049 that contains a photograph of the *Simtel MSDOS Archive*, October 1993 Edition, Walnut Creek CD-ROM, that has Simtel trade inscriptions. *See* Fed. R. Evid. 902(6)-(7).[5]

In its motion, PersonalWeb fails to identify, specifically, the textual portions of the aforementioned exhibits that allegedly are being offered for the truth of the matter asserted, yet seeks to exclude the entirety of each exhibit. The burden should not be placed on the Board to sort through the entirety of each exhibit and determine which portion of the exhibit PersonalWeb believes to be hearsay. Rather, PersonalWeb should have

[4] Fed. R. Evid. 901(b)(1). Testimony of a Witness with Knowledge.
Testimony that an item is what it is claimed to be.

[5] Fed. R. Evid. 902. Evidence that Is Self-Authenticating
The following items of evidence are self-authenticating; they require no extrinsic evidence of authenticity in order to be admitted:

(6) Newspapers and Periodicals. Printed material purporting to be a newspaper or periodical.

(7) Trade Inscriptions and the Like. An inscription, sign, tag, or label purporting to have been affixed in the course of business and indicating origin, ownership, or control.

identified, in its motion, the specific portions of the evidence and provided sufficient explanations as to why they constitute hearsay. Furthermore, PersonalWeb does not explain adequately why the declarations of Messrs. Sussell and Sadofsky do not provide the proper foundation and corroboration for the documents.

To the extent PersonalWeb relies upon the same arguments with respect to Kantor for excluding the documents, we have addressed those arguments above and determined that they are unavailing. We also agree with EMC that the documents concerning prior versions of Kantor are relevant, and not prejudicial or confusing, as alleged by PersonalWeb, because such circumstantial evidence provides context and corroboration for the witnesses' independent knowledge and recollection.

Furthermore, we are not persuaded that the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) should be excluded. As we discuss below in the next section, Messrs. Sussell and Sadofsky have sufficient personal knowledge and working experience to provide competent testimony to establish the publication and authentication of Kantor. The documents they cite serve to corroborate their independent knowledge and recollection.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude Exhibits 1047-1049, 1052-1055, 1074, 1075, 1080-1082, as well as the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) concerning those Exhibits.

Declarations of Messrs. Sussell and Sadofsky

PersonalWeb argues that the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) should be excluded as hearsay under Federal Rule of Evidence 801, and are inadmissible under Federal Rules of Evidence 802-807 for lack of foundation and personal knowledge, and Federal Rule of Evidence 702 as improper testimony, because the witnesses personally did not review Kantor (Ex. 1004) and Simtel (Ex. 1049) prior to the critical date. PO Mot. 8-10. PersonalWeb also argues that Messrs. Sussell and Sadofsky "are not qualified experts in the field." *Id*. at 10. PersonalWeb further alleges that Mr. Sadofsky's deposition (Ex. 2013, 30, 66) should be excluded, as it was responsive to a leading question and non-responsive to the question. *Id*.

EMC responds that the testimony of Messrs. Sussell and Sadofsky should not be excluded because their testimony is based on their own personal knowledge and recollection, and the documents they cite serve to corroborate their independent knowledge and recollection. Opp. 11-12. EMC further explains that the witnesses have described thoroughly the underlying facts, and, therefore, the testimony should be admitted as relevant under Federal Rules of Evidence 401-402, supported by personal knowledge and foundation under Federal Rule of Evidence 602, and proper opinion testimony under Federal Rules of Evidence 701-703. *Id*. We find EMC's contentions have merit.

PersonalWeb's arguments rest on the erroneous premise that EMC's witnesses must have reviewed Kantor or Simtel personally prior to the critical date in order to provide competent testimony regarding Kantor or Simtel. As discussed previously, it is well settled that it is not necessary for the witnesses to have reviewed the reference personally prior to the critical date in order to establish publication. *See, e.g., Wyer*, 655 F.2d at 226.

Although Messrs. Sussell and Sadofsky are not experts related to the claimed subject matter of the ’096 Patent, each witness nevertheless has sufficient personal knowledge and working experience to provide competent testimony. *See Hall*, 781 F.2d at 899. Mr. Sussell was the co-owner and system operator of the Invention Factory Bulletin Board System from 1983 to 1996. Ex. 1050 ¶ 3. Mr. Sussell’s testimony is based on his personal knowledge of the relevant facts related to the Invention Factory Bulletin Board System and Kantor. *Id.* at ¶ 2. Notably, Dr. Kantor specifically thanked Mr. Sussell in his user manual for hosting Dr. Kantor’s software FWKCS and for Mr. Sussell’s role in its development. Ex. 1004, 3 (“To Michael Sussell, sysop of The Invention Factory (R), home board for the support of FWKCS, for bringing the problem of duplicate files to my attention and for his help in testing . . . .”); *id.* at 6 (“When Michael Sussell, sysop of The Invention Factory (R) in New York, brought to my attention the problem of duplicate files with different names, these concepts provided valuable insight into how one might proceed.”).

Mr. Sadofsky is a technology archivist and software historian, and works “for the Internet Archive, a non-profit digital library offering free universal access to books, movies, and music, as well as 342 billion archived webpages available through the Wayback Machine service.” Ex. 1078 ¶ 3.

Mr. Sadofsky also "directed the film, *The BBS Documentary*, an eight-episode documentary about the subculture born from the creation of the [Bulletin Board System]." *Id*. at ¶ 4. Mr. Sadofsky's testimony is based on his personal knowledge of the relevant facts related to Kantor and the "1993 archived" version of Kantor. *Id*. at ¶ 2; Ex. 1088 ¶ 2. For example, Mr. Sadofsky personally verified the authenticity of Kantor by comparing it with the "1993 archived" version, and determined that Kantor—version 1.22, the version relied upon by EMC (Ex. 1004)—is identical to the "1993 archived" version. Ex. 1086 ¶¶ 14-15.

Upon review of the evidence on the record, we agree with EMC that both Messrs. Sussell and Sadofsky have disclosed sufficient underlying facts to support their testimony. For instance, the computer-generated timestamp—August 10, 1993, 1:22 AM—associated with the "1993 archived" version of Kantor corroborates their testimony regarding Kantor's existence as of August 10, 1993. Ex. 1078 ¶¶ 14-15; Ex.1088 ¶¶ 10-11; Ex. 2014 ¶ 5.

As to Mr. Sadofsky's deposition (Ex. 2013, 30, 66), PersonalWeb does not explain sufficiently why that testimony should be excluded. PO Mot. 11. Moreover, Mr. Sadofsky's deposition (Ex. 2013, 30, 66) is consistent with his direct testimony (Ex. 1078 ¶¶ 14-16), and, therefore, it would not prejudice PersonalWeb even if such evidence is not excluded.

For the foregoing reasons, PersonalWeb has not presented a sufficient basis to exclude the declarations of Messrs. Sussell and Sadofsky (Exs. 1050, 1078, 1088) and Mr. Sadofsky's deposition (Ex. 2013, 30, 66).

Clark's Rebuttal Declaration

PersonalWeb asserts that Dr. Clark's rebuttal declaration (Ex. 1089) should be excluded, because it is irrelevant, prejudicial, and confusing, as well as beyond the scope of this proceeding. PO Mot. 10-15. In support of its assertion, PersonalWeb advances several arguments. *Id*.

First, PersonalWeb argues that Dr. Clark's rebuttal declaration cites to references that do not serve as the basis of a ground of unpatentability instituted in this proceeding, i.e., Browne and Langer. *Id*. at 10-11. EMC counters that Dr. Clark's statements referencing those references were offered in response to PersonalWeb's argument that one with ordinary skill in the art would not have modified Kantor. Opp. 12 (citing PO Resp. 35-36; Ex. 2017 ¶¶ 60-62). According to EMC, those statements are relevant to the instituted grounds of unpatentability and confirm that the use of hash-based identifiers to identify files was well known in the art at the time of invention. *Id*. We agree with EMC that Dr. Clark's statements are proper rebuttal evidence submitted in response to PersonalWeb's arguments. Those references were cited merely to show the knowledge level of a person with ordinary skill in the art. *See Randall Mfg. v. Rea*, 733 F.3d 1355, 1362 (Fed. Cir. 2013) (When considering whether a claimed invention would have been obvious, "the knowledge of [an ordinarily skilled] artisan is part of the store of public knowledge that must be consulted."). Such evidence does not change the combination that formed the basis of the grounds of unpatentability based on obviousness instituted in this proceeding. *Id*.; *see* *also In re Donohue*, 766 F.2d 531, 534 (Fed. Cir. 1985). Accordingly, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude Dr. Clark's rebuttal declaration.

Second, PersonalWeb contends that the "could" and "might" statements in Dr. Clark's rebuttal declaration should be excluded, because those statements are irrelevant, prejudicial, confusing, lacking foundation, and beyond the scope of this proceeding. PO Mot. 11. In response, EMC contends that the statements in Dr. Clark's rebuttal declaration were offered in response to PersonalWeb's arguments. Opp. 13 (citing *e.g.*, PO Resp. 27; Ex. 2017 ¶ 49). Having reviewed PersonalWeb's patent owner response and Dr. Clark's rebuttal declaration, we determine that Dr. Clark's testimony is reasonable rebuttal evidence in light of PersonalWeb's arguments. Furthermore, PersonalWeb's arguments concerning Dr. Clark's statements affect the weight to be given by us to Dr. Clark's testimony in deciding whether the instituted grounds of unpatentability render the challenged claimed unpatentable. When weighing evidence, we are capable of determining whether the prior art references anticipate or render obvious the challenged claims without being confused, misled, or prejudiced by Dr. Clark's testimony. Thus, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude any portions of Dr. Clark's rebuttal declaration.

Third, PersonalWeb submits that Dr. Clark's rebuttal declaration of what Kantor's software allegedly could do is irrelevant because an *inter partes* review is limited to printed publications and patents. *Id*. at 11-12.

EMC counters that Dr. Clark's rebuttal declaration (Ex. 1089 ¶¶ 26) regarding what Kantor's software "could" do is relevant and admissible. Opp. 13. EMC points out that Dr. Clark's rebuttal declaration merely explains "what a person of skill in the art, reading the Kantor reference, would [have understood] the reference to disclose about Kantor's software." *Id.* (citing Ex. 1089 ¶ 26). We agree with EMC that Dr. Clark is not offering opinions on how Kantor's software might operate but, rather, on what it discloses or suggests. *Id.* Thus, we conclude it is relevant and we are not persuaded that PersonalWeb has presented a sufficient basis to exclude this portion of Dr. Clark's rebuttal declaration.

Fourth, PersonalWeb also submits that Dr. Clark's rebuttal declaration includes new obviousness allegations not presented previously with the petition. *Id.* at 12-13. In response, EMC contends that the statements in Dr. Clark's rebuttal declaration were offered in response to PersonalWeb's arguments. Opp. 14 (citing *e.g.*, PO Resp. 3-18, 27; Ex. 2017 ¶¶ 21-37, 60-62). Having reviewed PersonalWeb's patent owner response and Dr. Clark's rebuttal declaration, we determine that Dr. Clark's testimony is reasonable rebuttal evidence in light of PersonalWeb's arguments. Thus, we are not persuaded that PersonalWeb has presented a sufficient basis to exclude any portions of Dr. Clark's rebuttal declaration.

Finally, PersonalWeb contends that Dr. Clark's rebuttal declaration contradicts his prior deposition. PO Mot. 13-15. We are not persuaded by PersonalWeb's arguments. Rather, we agree with EMC that Dr. Clark's rebuttal testimony is consistent with his earlier testimony. Opp. 14-15.

PersonalWeb argues that Dr. Clark has changed his testimony on what part of Kantor is a "data item" (PO Mot. 13-14), but we agree with EMC that Dr. Clark has focused on the inner files of the zip file as the relevant portion. *Compare* Ex. 1009 ¶¶ 86, 100 *with* Ex. 1089 ¶¶ 7-10. Additionally, PersonalWeb argues that Dr. Clark has changed his testimony with respect to whether a "sequence" can have gaps (PO Mot. 14), where EMC argues that his testimony is consistent. Opp. 14. We do not discern that Dr. Clark's answer to a question related to "a sequence of *people*" (Ex. 2016, 94-98) contradicts with Dr. Clark's rebuttal testimony on "a sequence of *bits*" of a data item (Ex. 1089 ¶ 28). Dr. Clark in the prior deposition also testified that there are examples of sequences with intervening gaps including Fibonacci sequences, random sequences, odd sequences, and even sequences. Opp. 15 (citing Ex. 2016, 191-193).

In addition, Dr. Clark's rebuttal testimony that "zipfiles are not *always* compressed," and the inner files of a zip file may be *uncompressed* (Ex. 1089 ¶¶ 9-11), is consistent with his earlier testimony that the inner files of a zip file are compressed *typically* (Ex. 2016, 55, 59, 66-67). Moreover, Dr. Clark's testimony is reasonable rebuttal evidence in light of the evidence submitted by PersonalWeb. Dr. Clark merely points out in his rebuttal declaration that PersonalWeb's evidence also shows that zip files are not *always* compressed. Ex. 1089 ¶ 9 (citing Ex. 2004, 3 (the zip file format defines seven compression methods which include "Compression method 0" that does not compress the file); Ex. 1083, 262 (Dr. Dewar agrees that "the zipfile standard allows for uncompressed files.")).

For the foregoing reasons, we decline to exclude Dr. Clark's rebuttal declaration (Ex. 1089).

III. CONCLUSION

EMC has met its burden of proof, by a preponderance of the evidence, in showing that claims 1, 2, 81, and 83 of the '096 Patent are unpatentable based on the following ground of unpatentability:

Claims	Basis	References
1, 2, 81, and 83	§ 103(a)	Kantor and Satyanarayanan

IV. ORDER

In consideration of the foregoing, it is

ORDERED that claims 1, 2, 81, and 83 of the '096 Patent are held unpatentable;

FURTHER ORDERED that EMC's Motion to Exclude Evidence is *dismissed*;

FURTHER ORDERED that PersonalWeb's Motion to Exclude Evidence is *denied*; and

FURTHER ORDERED that, because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Peter M. Dichiara, Esq.
David L. Cavanaugh, Esq.
WILMER CUTLER PICKERING HALE & DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com

PATENT OWNER:

Joseph A. Rhoa, Esq.
Updeep S. Gill, Esq.
NIXON & VANDERHYE P.C.
jar@nixonvan.com
usg@nixonvan.com

US005978791A

United States Patent [19]

Farber et al.

[11] **Patent Number: 5,978,791**

[45] **Date of Patent: Nov. 2, 1999**

[54] **DATA PROCESSING SYSTEM USING SUBSTANTIALLY UNIQUE IDENTIFIERS TO IDENTIFY DATA ITEMS, WHEREBY IDENTICAL DATA ITEMS HAVE THE SAME IDENTIFIERS**

[75] Inventors: **David A. Farber**, Ojai, Calif.; **Ronald D. Lachman**, Northbrook, Ill.

[73] Assignee: **Kinetech, Inc.**, Northbrook, Ill.

[21] Appl. No.: **08/960,079**

[22] Filed: **Oct. 24, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/425,160, Apr. 11, 1995, abandoned.

[51] **Int. Cl.**^{6} **G06F 17/30**

[52] **U.S. Cl.** **707/2**; 707/1; 707/200

[58] **Field of Search** 707/2, 1, 200

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,668,647	6/1972	Evangelisti et al.	340/172.5
4,215,402	7/1980	Mitchell et al.	364/200
4,290,105	9/1981	Cichelli et al.	364/200
4,376,299	3/1983	Rivest	364/900
4,405,829	9/1983	Rivest et al.	178/22.1
4,412,285	10/1983	Neches et al.	364/200
4,414,624	11/1983	Summer, Jr. et al.	364/200
4,441,155	4/1984	Fletcher et al.	364/200
4,464,713	8/1984	Benhase et al.	364/200
4,490,782	12/1984	Dixon et al.	364/200
4,571,700	2/1986	Emry, Jr. et al.	364/900
4,577,293	3/1986	Matick et al.	365/189
4,642,793	2/1987	Meaden	364/900
4,675,810	6/1987	Gruner et al.	364/200
4,691,299	9/1987	Rivest et al.	365/185
4,725,945	2/1988	Kronstadt et al.	364/200
4,773,039	9/1988	Zamora	364/900
4,887,235	12/1989	Holloway et al.	364/900
4,888,681	12/1989	Barnes et al.	364/200
4,922,414	5/1990	Holloway et al.	364/200
4,972,367	11/1990	Burke	364/900
5,007,658	4/1991	Bendert et al.	395/600
5,025,421	6/1991	Cho	365/230.05
5,050,074	9/1991	Marca	364/200
5,050,212	9/1991	Dyson	380/25
5,057,837	10/1991	Colwell et al.	341/55
5,129,081	7/1992	Kobayashi et al.	395/600
5,129,082	7/1992	Tirfing et al.	395/600
5,144,667	9/1992	Pogue, Jr. et al.	380/45
5,179,680	1/1993	Colwell et al.	395/425
5,202,982	4/1993	Gramlich et al.	395/600
5,208,858	5/1993	Vollert et al.	380/43
5,276,901	1/1994	Howell et al.	395/800
5,301,286	4/1994	Rajani	395/400
5,301,316	4/1994	Hamilton et al.	395/600
5,343,527	8/1994	Moore	380/4
5,357,623	10/1994	Megory-Cohen	395/425
5,384,565	1/1995	Cannon	340/825.44
5,404,508	4/1995	Konrad et al.	395/600

OTHER PUBLICATIONS

Witold Litwin et al, Linear Hashing for Distributed Files, ACM SIGMOD, May, 1993 pp. 327–336.

Ming–Ling Lo, et al, On Optimal Processor Allocation to Support Pipelined Hash Joins, ACM SIGMOD, pp. 69–78, May 1993.

Thomas A. Berson, Differential Cryptanalysis Mod 2^{32} with Applications to MD5, pp. 69–81, 1992.

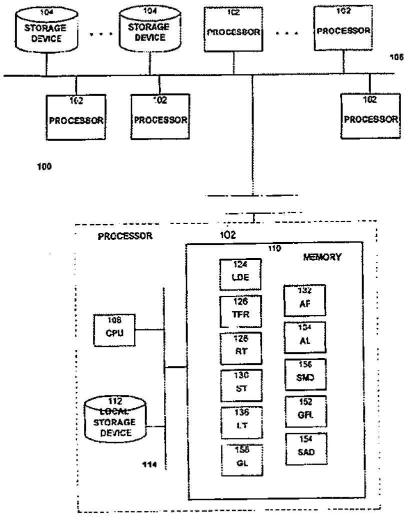
(List continued on next page.)

Primary Examiner—Paul V. Kulik
Assistant Examiner—Jean R. Homere
Attorney, Agent, or Firm—Pillsbury Madison & Sutro LLP

[57] **ABSTRACT**

In a data processing system, a mechanism identifies data items by substantially unique identifiers which depend on all of the data in the data items and only on the data in the data items. The system also determines whether a particular data item is present in the database by examining the identifiers of the plurality of data items.

48 Claims, 31 Drawing Sheets



OTHER PUBLICATIONS

William Perrizo, et al., Distributed Join Processing Performance Evaluation, 1994. Twenty–Seventh Hawaii International Conference on System Sciences, vol. II, pp. 236–244.

A concurrency Control Mechanism based on Extendible Hashing for Main Memory Database Systems, Vijay Kumar, pp. 109–113, ACM, vol. 3, 1989.

Birgit Pfitzmann, Sorting Out Signature Schemes, Nov. 1993, 1st Conf. Computer & Comm. Security '93 pp. 74–85.

Bert dem Boer, et al., Collisions for the compression function of MD_5 pp. 292–304, 1994.

Sakti Pramanik, et al., Multi–Directory Hashing, 1993, Info. Sys., vol. 18, No. 1, pp. 63–74.

Murlidhar Koushik, Dynamic Hashing With Distributed Overflow Space: A File Organization With Good Insertion Performance, 1993, Info. Sys., vol. 18, No. 5, pp. 299–317.

Witold Litwin, et al., LH*–Linear Hashing for Distributed Files, HP Labs Tech. Report No. HPL–93–21 Jun. 1993 pp. 1–22.

Yuliang Zheng, et al., HAVAL — A One–Way Hashing Algorithm with Variable Length of Output (Extended Abstract), pp. 83–105, Advances in Cryptology, AUSCRIPT '92, 1992.

Chris Charnes and Josef Pieprzky, Linear Nonequivalence versus Nonlinearity, Pieprzky, pp. 156–164, 1993.

Zhiyu Tian, et al., A New Hashing Function: Statistical Behaviour and Algorithm, pp. 3–13, SIGIR Forum, 1993.

G. L. Friedman, Digital Camera With Apparatus For Authentication of Images Produced From an Image File, NASA Case No. NPO–19108–1–CU, Serial No. 08/159,980, Nov. 24, 1993.

H. Goodman, Feb. 9, 1994 Ada, Object–Oriented Techniques, and Concurrency in Teaching Data Sructures and File Management Report Documentation P. AD–A275 385 — 94–04277.

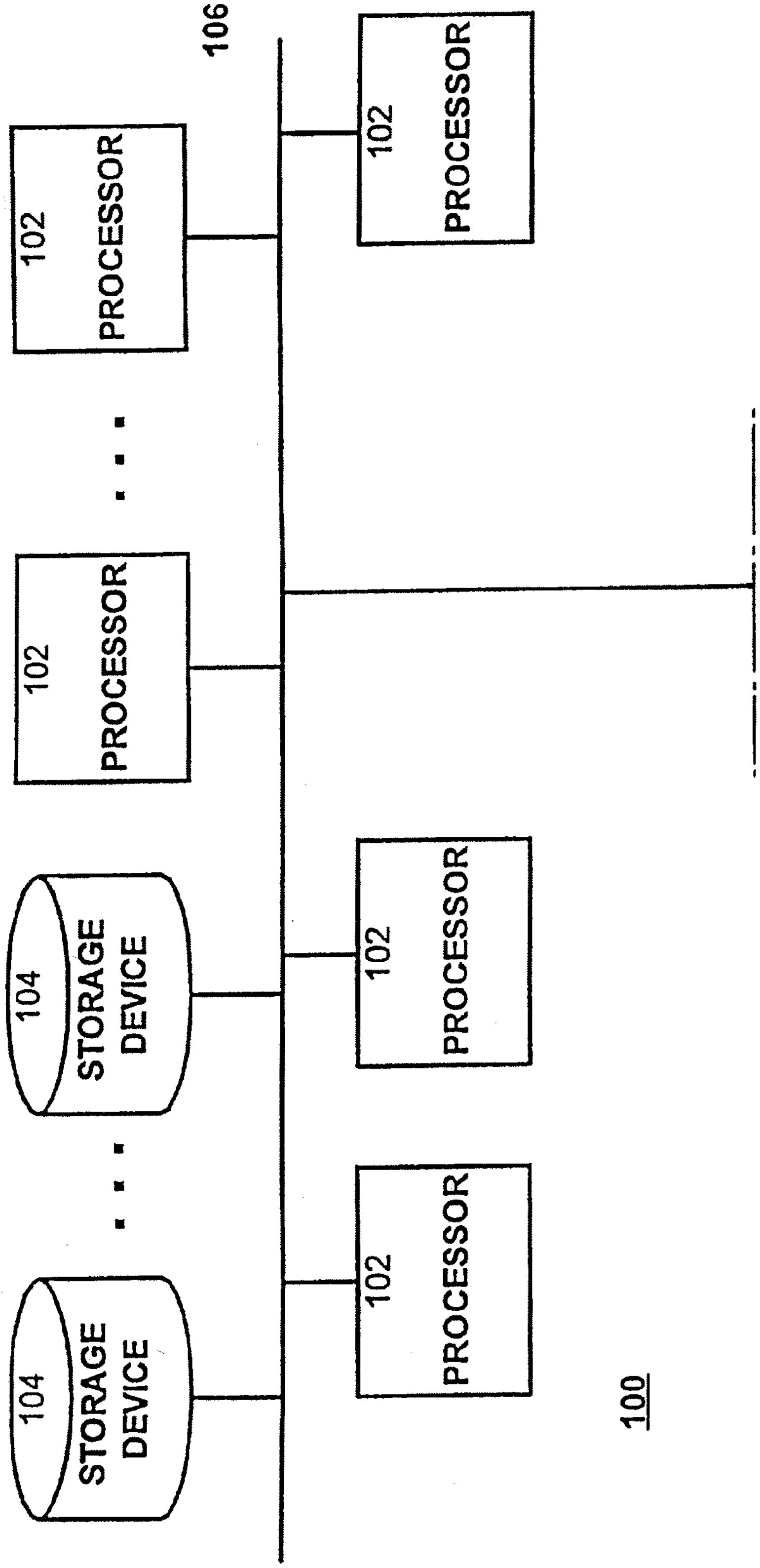
Advances in Cryptology–EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings.

Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, vol. 22, Issue 2, Jun. 1993.

Advances in Cryptology–AUSCRYPT '92 — Workshop on the Theory and Application of Cryptographic Techniques Gold Coast, Queensland, Australia Dec. 13–16, 1992 Proceedings.

Search Report dated Jun. 24, 1996.

FIG. 1(a)


106
102
PROCESSOR
102
PROCESSOR
102
PROCESSOR
104
STORAGE DEVICE
102
PROCESSOR
104
STORAGE DEVICE
102
PROCESSOR
100

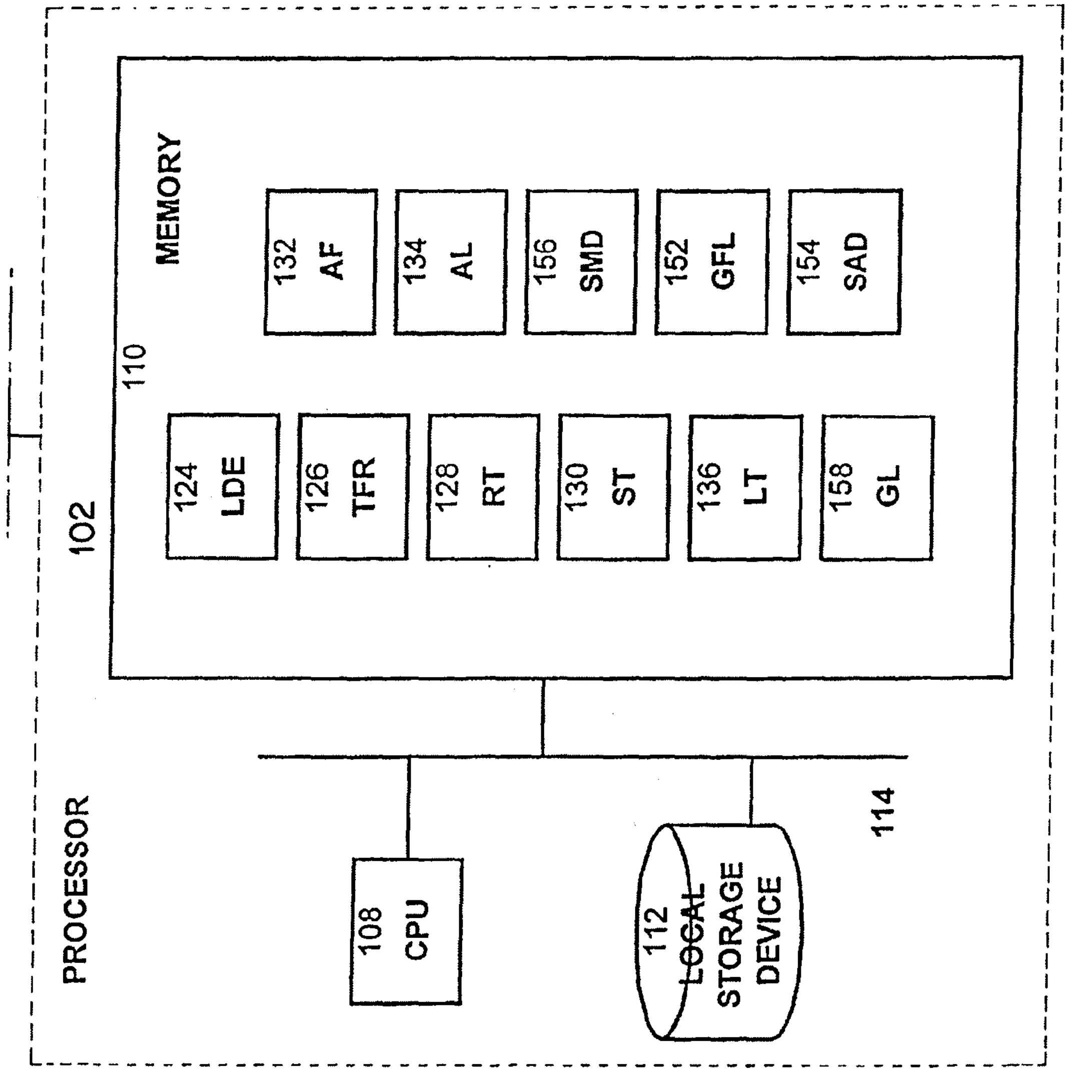

PROCESSOR
102
MEMORY
110
132 AF
134 AL
156 SMD
152 GFL
154 SAD
124 LDE
126 TFR
128 RT
130 ST
136 LT
158 GL
108 CPU
112 LOCAL STORAGE DEVICE
114

FIG. 1(b)

FIG. 2

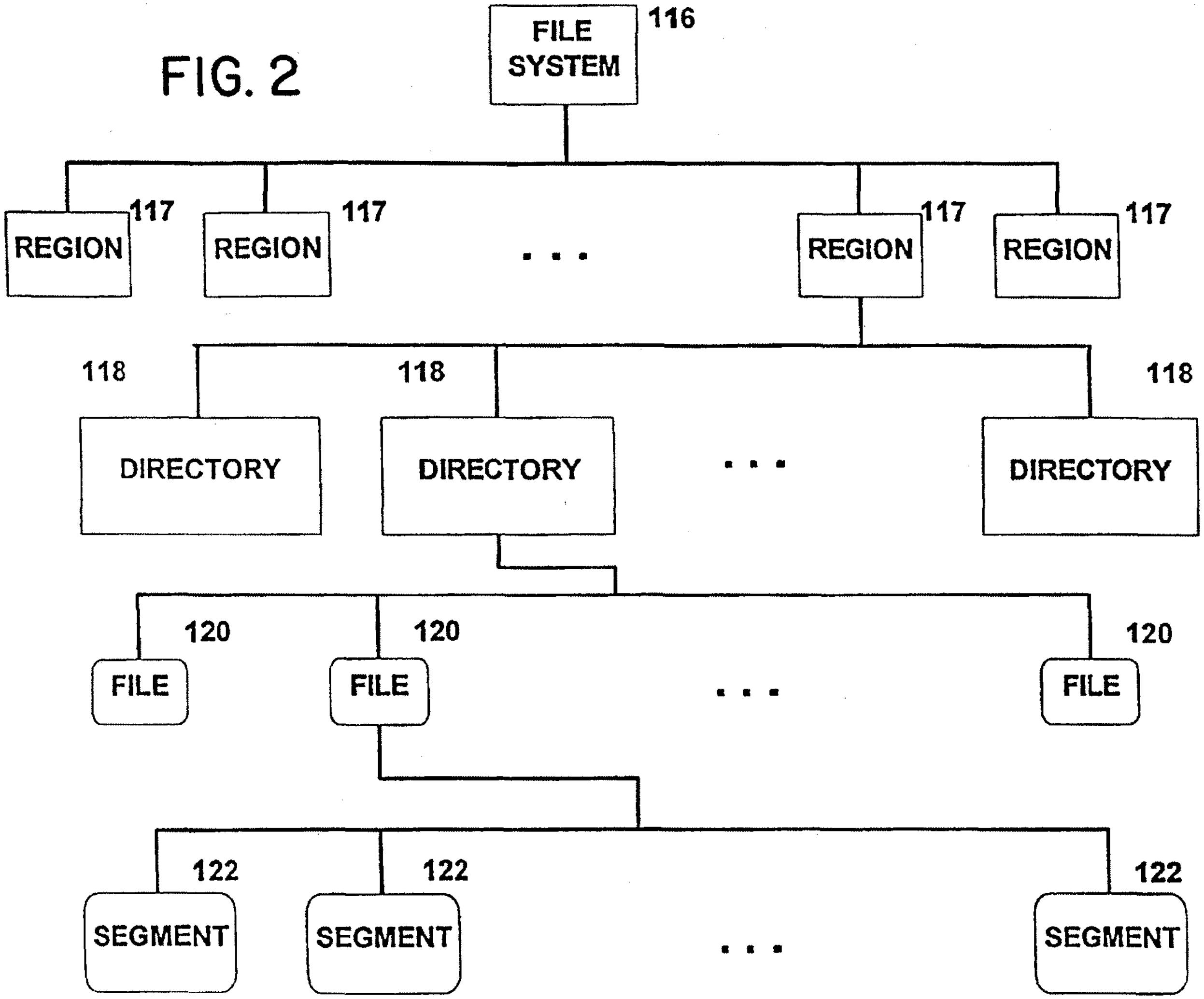

FILE SYSTEM
116
REGION
117
REGION
117
. . .
REGION
117
REGION
117
118
DIRECTORY
118
DIRECTORY
. . .
118
DIRECTORY
FILE
120
FILE
120
. . .
FILE
120
SEGMENT
122
SEGMENT
122
. . .
SEGMENT
122

FIG. 3

138

Region ID
Pathname
True Name
Type
File ID
Time of last access
Time of last modification
Safe flag
Lock flag
Size
Owner

FIG. 4

140

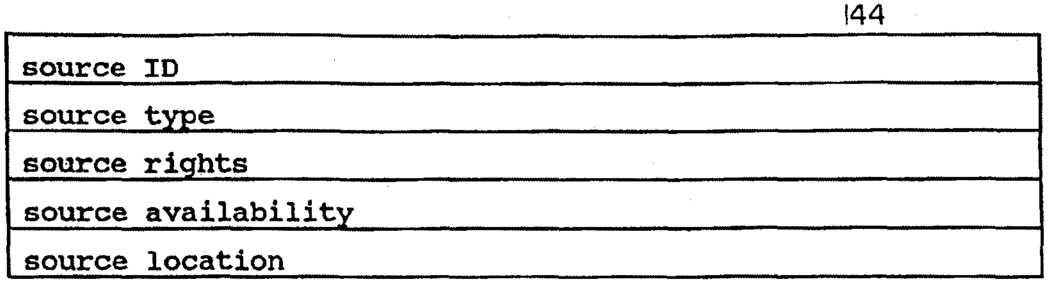
True Name
File ID
Compressed File ID
Source IDs
Dependent processors
Use count
Time of last access
Expiration
Grooming delete count

142

Region ID
Region file system
Region pathname
Region status
Mirror processor(s)
Mirror duplication count
Policy

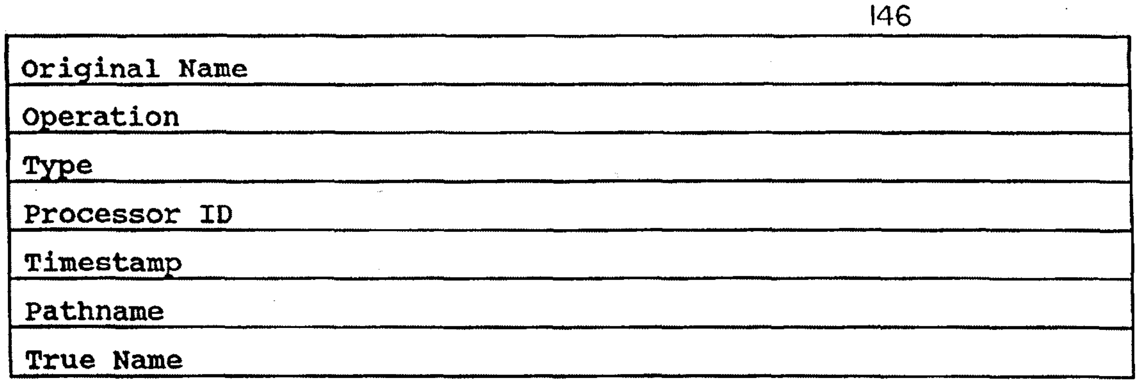
FIG. 5

FIG. 6



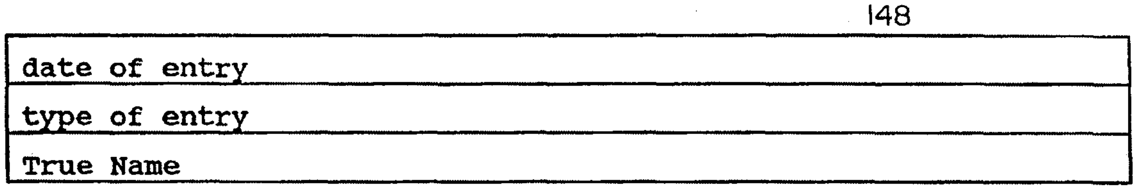
source ID
source type
source rights
source availability
source location

FIG. 7



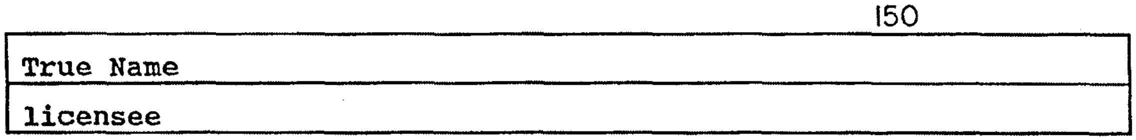
Original Name
Operation
Type
Processor ID
Timestamp
Pathname
True Name

FIG. 8



date of entry
type of entry
True Name

FIG. 9



True Name
licensee

FIG. 10(a)

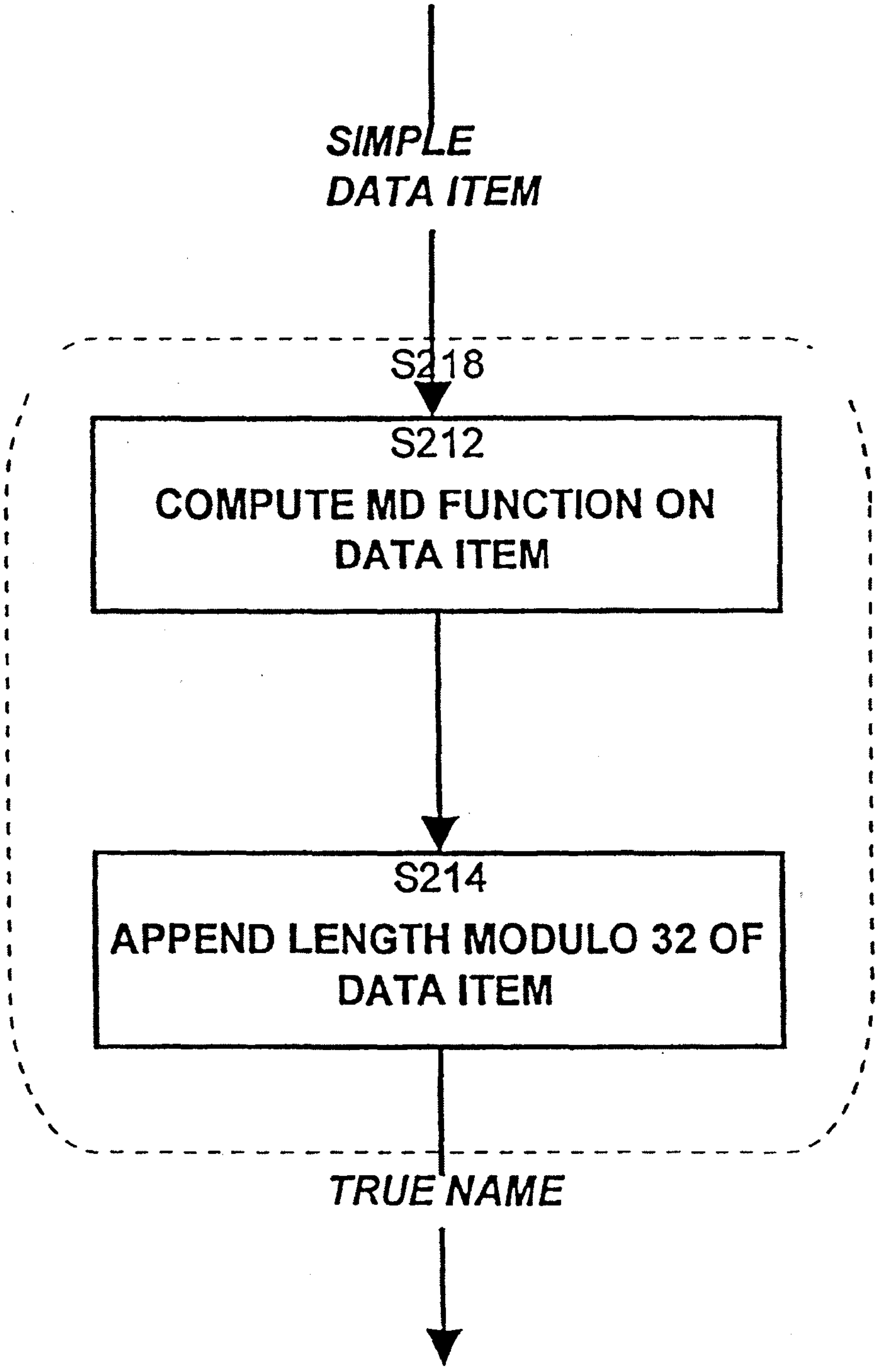

SIMPLE
DATA ITEM
S218
S212
COMPUTE MD FUNCTION ON DATA ITEM
S214
APPEND LENGTH MODULO 32 OF DATA ITEM
TRUE NAME

FIG. 10(b)

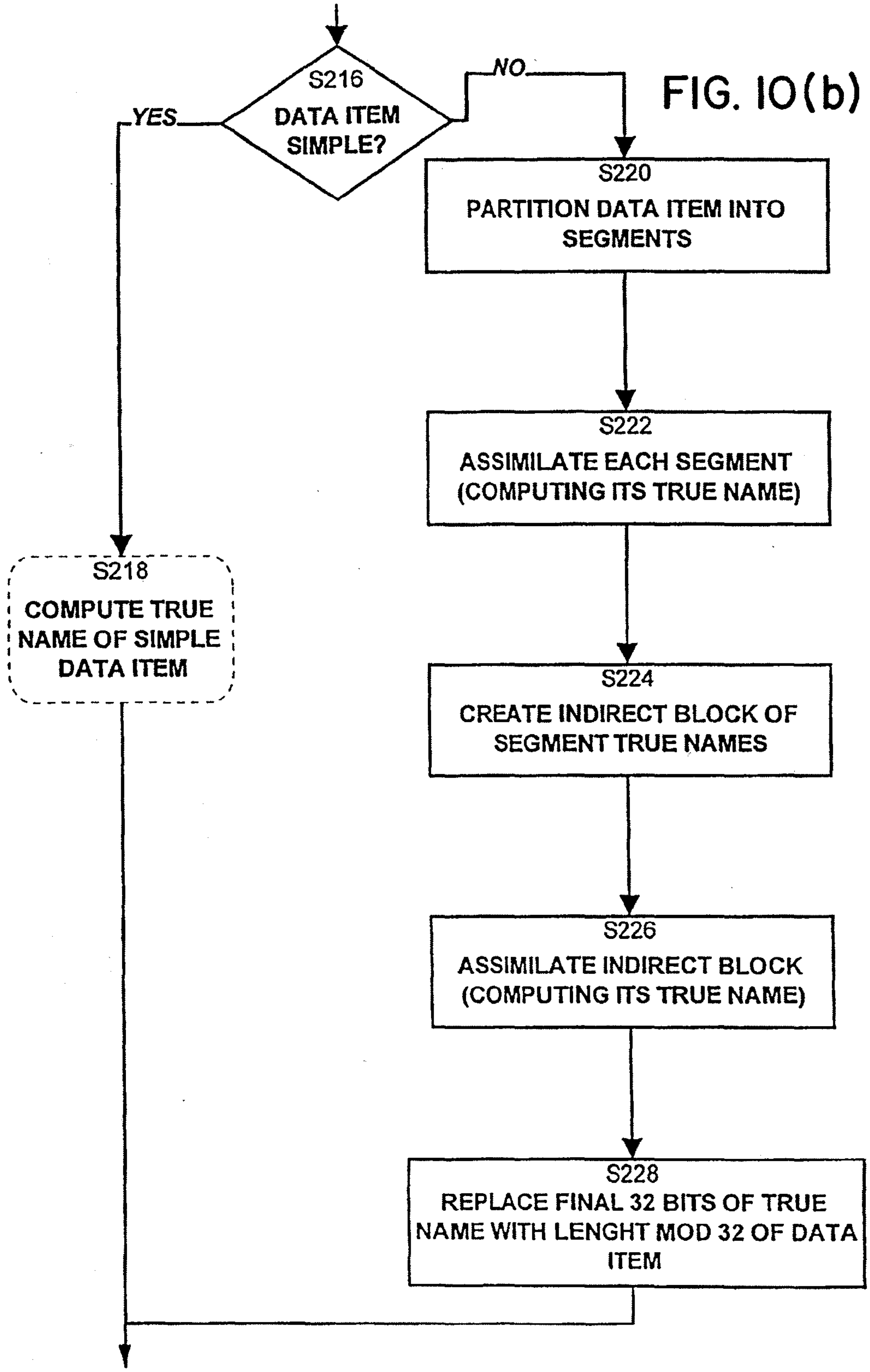

S216
DATA ITEM SIMPLE?
YES
NO
S220
PARTITION DATA ITEM INTO SEGMENTS
S222
ASSIMILATE EACH SEGMENT (COMPUTING ITS TRUE NAME)
S218
COMPUTE TRUE NAME OF SIMPLE DATA ITEM
S224
CREATE INDIRECT BLOCK OF SEGMENT TRUE NAMES
S226
ASSIMILATE INDIRECT BLOCK (COMPUTING ITS TRUE NAME)
S228
REPLACE FINAL 32 BITS OF TRUE NAME WITH LENGHT MOD 32 OF DATA ITEM

FIG. 11

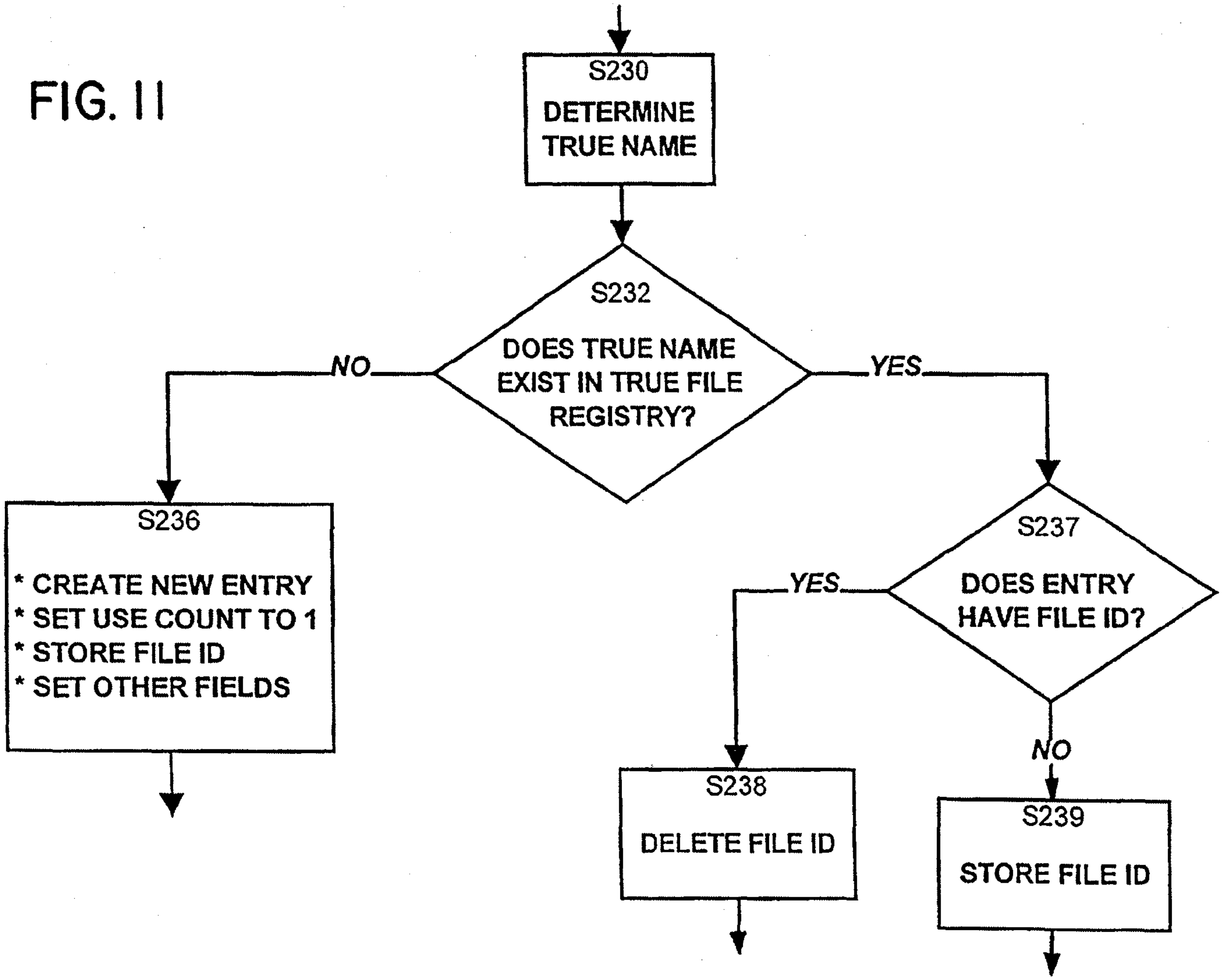

S230
DETERMINE TRUE NAME
S232
DOES TRUE NAME EXIST IN TRUE FILE REGISTRY?
NO
YES
S236
* CREATE NEW ENTRY
* SET USE COUNT TO 1
* STORE FILE ID
* SET OTHER FIELDS
S237
DOES ENTRY HAVE FILE ID?
YES
NO
S238
DELETE FILE ID
S239
STORE FILE ID

FIG. 12

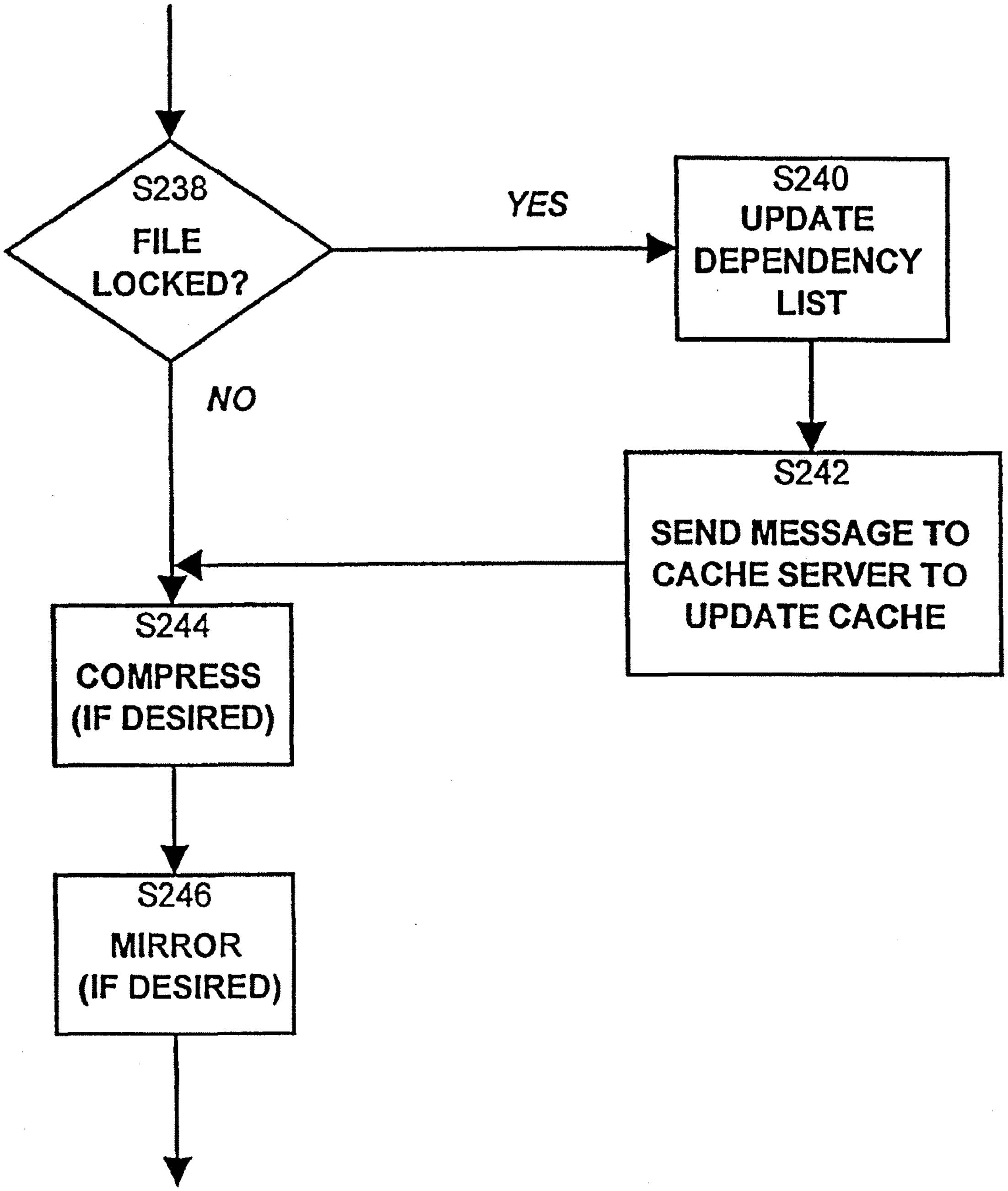
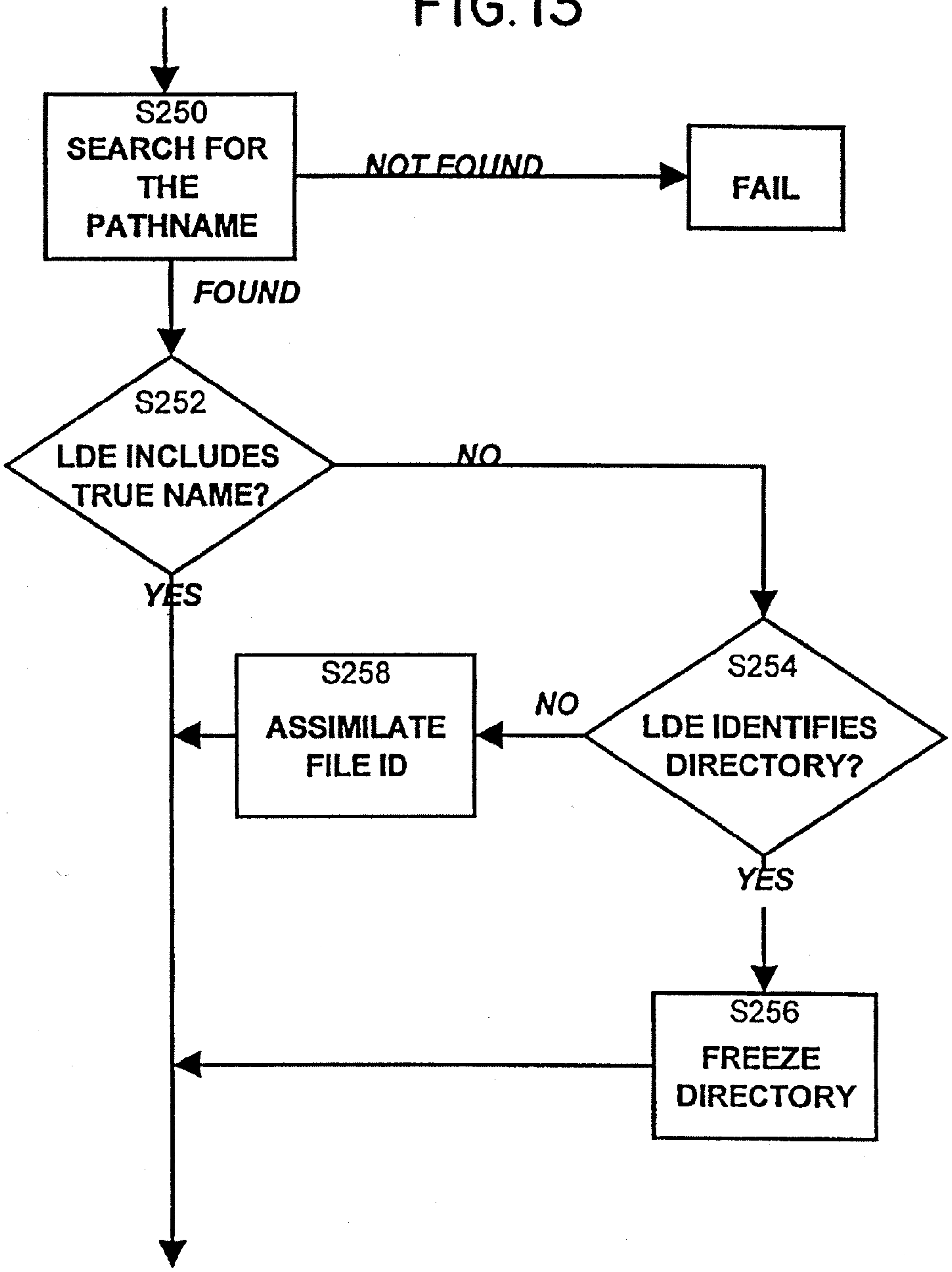

S238
FILE LOCKED?
YES
NO
S240
UPDATE DEPENDENCY LIST
S242
SEND MESSAGE TO CACHE SERVER TO UPDATE CACHE
S244
COMPRESS (IF DESIRED)
S246
MIRROR (IF DESIRED)

FIG. 13


S250
SEARCH FOR THE PATHNAME
NOT FOUND
FAIL
FOUND
S252
LDE INCLUDES TRUE NAME?
NO
YES
S254
LDE IDENTIFIES DIRECTORY?
NO
S258
ASSIMILATE FILE ID
YES
S256
FREEZE DIRECTORY

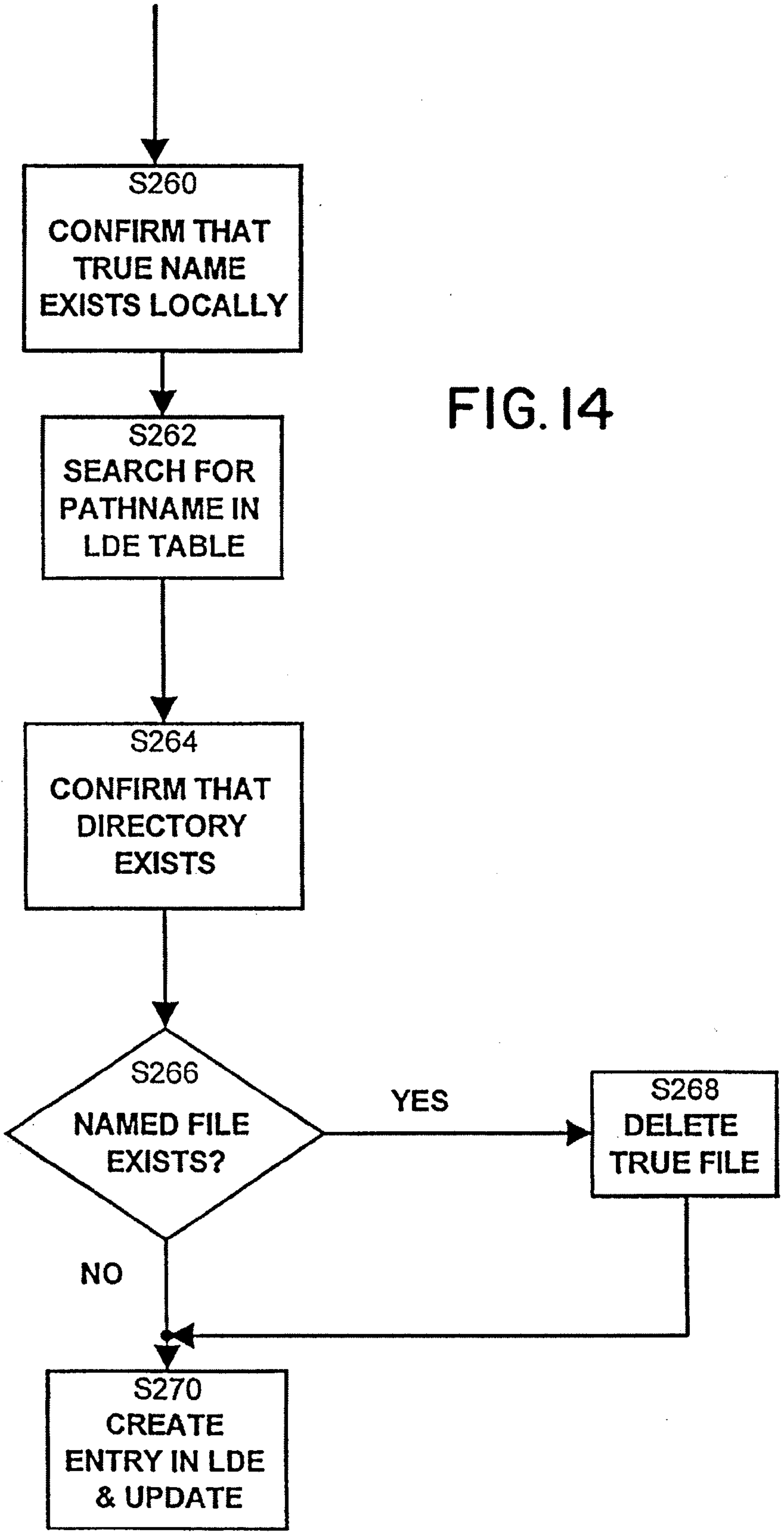

S260
CONFIRM THAT TRUE NAME EXISTS LOCALLY
S262
SEARCH FOR PATHNAME IN LDE TABLE
S264
CONFIRM THAT DIRECTORY EXISTS
S266
NAMED FILE EXISTS?
YES
S268
DELETE TRUE FILE
NO
S270
CREATE ENTRY IN LDE & UPDATE

FIG. 14

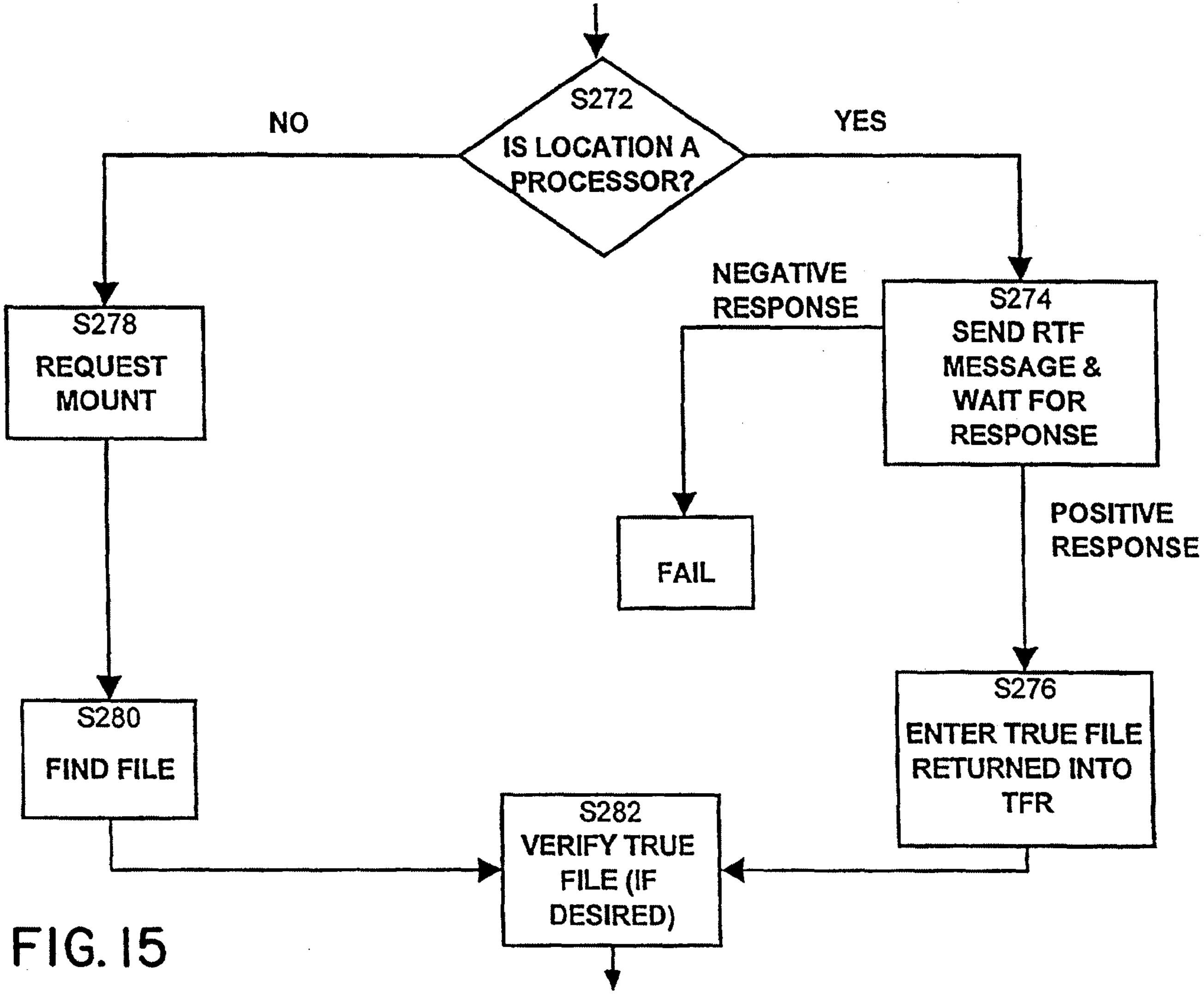

S272
IS LOCATION A PROCESSOR?
NO
YES
S278
REQUEST MOUNT
S274
SEND RTF MESSAGE & WAIT FOR RESPONSE
NEGATIVE RESPONSE
FAIL
POSITIVE RESPONSE
S280
FIND FILE
S276
ENTER TRUE FILE RETURNED INTO TFR
S282
VERIFY TRUE FILE (IF DESIRED)

FIG. 15

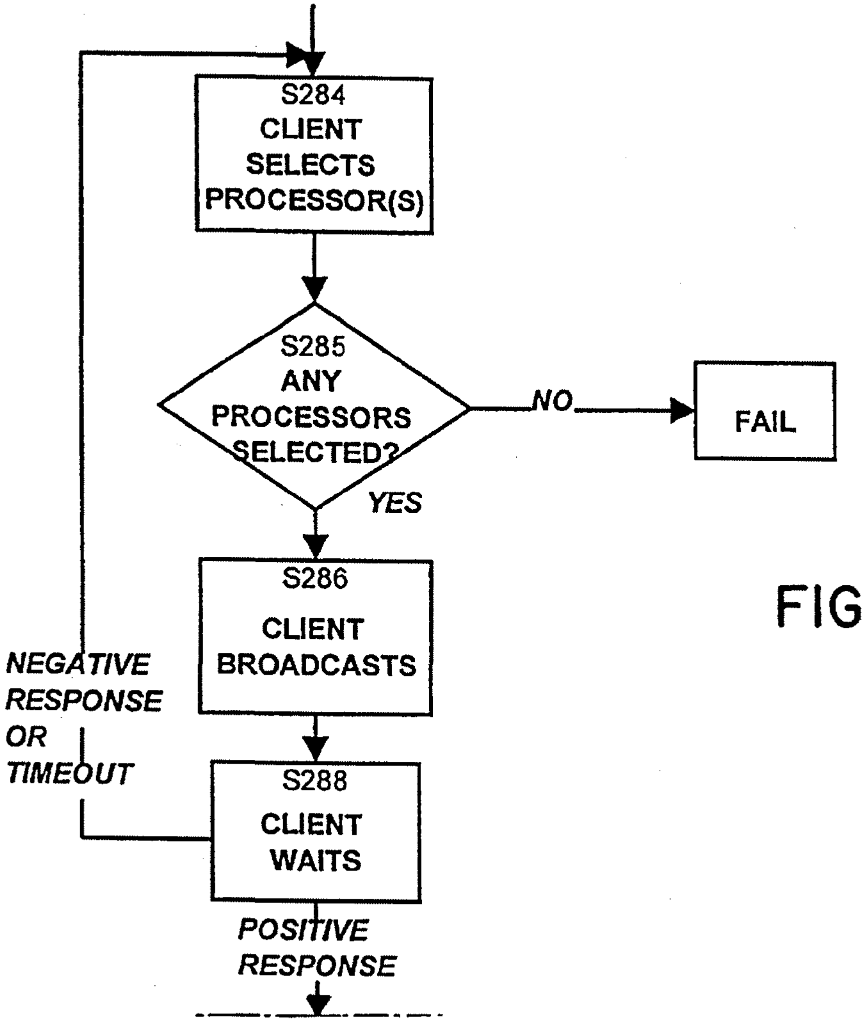

S284
CLIENT SELECTS PROCESSOR(S)
S285
ANY PROCESSORS SELECTED?
NO
FAIL
YES
S286
CLIENT BROADCASTS
S288
CLIENT WAITS
NEGATIVE RESPONSE OR TIMEOUT
POSITIVE RESPONSE

FIG. 16(a)

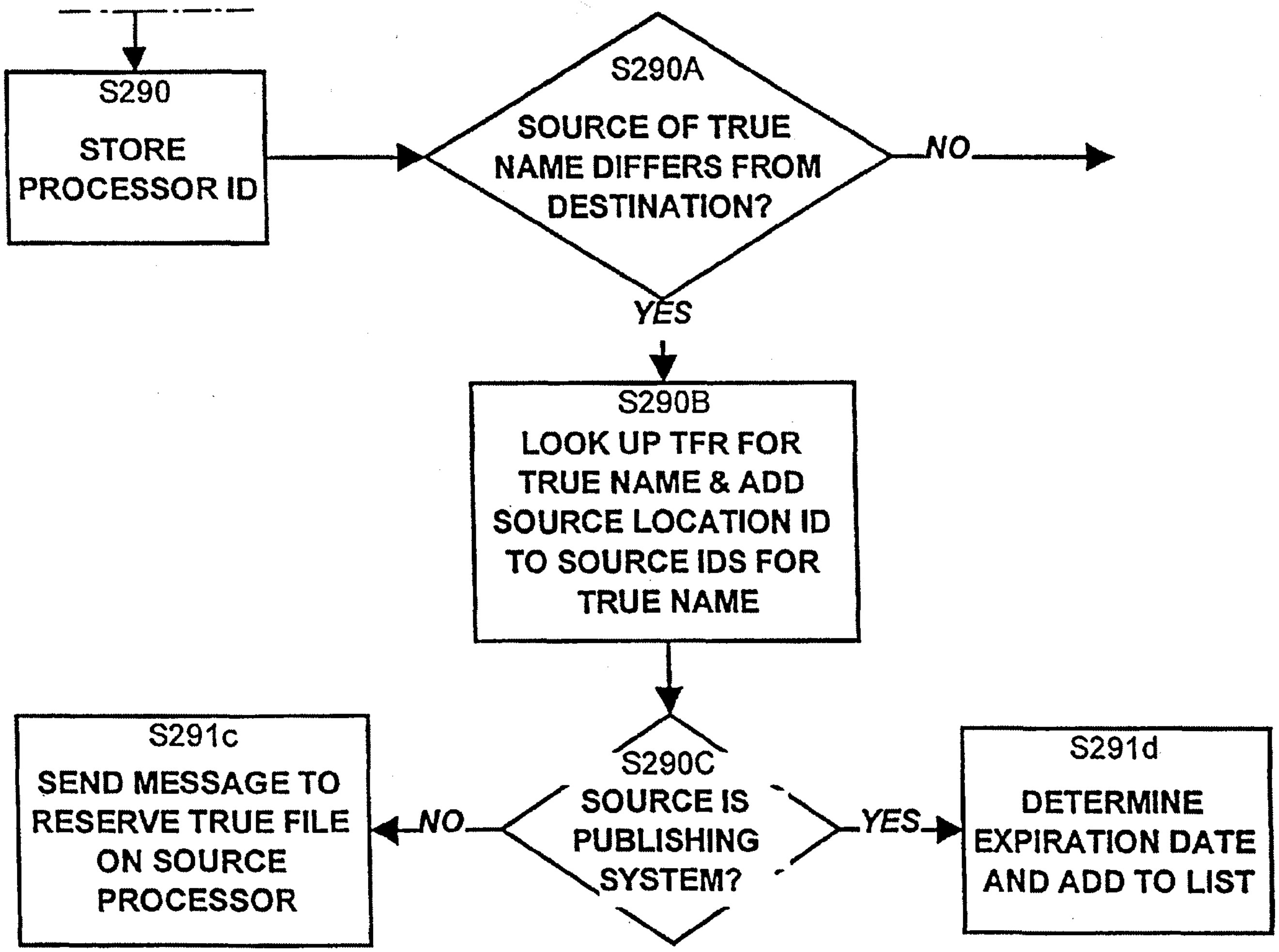

S290
STORE PROCESSOR ID
S290A
SOURCE OF TRUE NAME DIFFERS FROM DESTINATION?
NO
YES
S290B
LOOK UP TFR FOR TRUE NAME & ADD SOURCE LOCATION ID TO SOURCE IDS FOR TRUE NAME
S290C
SOURCE IS PUBLISHING SYSTEM?
NO
YES
S291c
SEND MESSAGE TO RESERVE TRUE FILE ON SOURCE PROCESSOR
S291d
DETERMINE EXPIRATION DATE AND ADD TO LIST

FIG. 16(b)

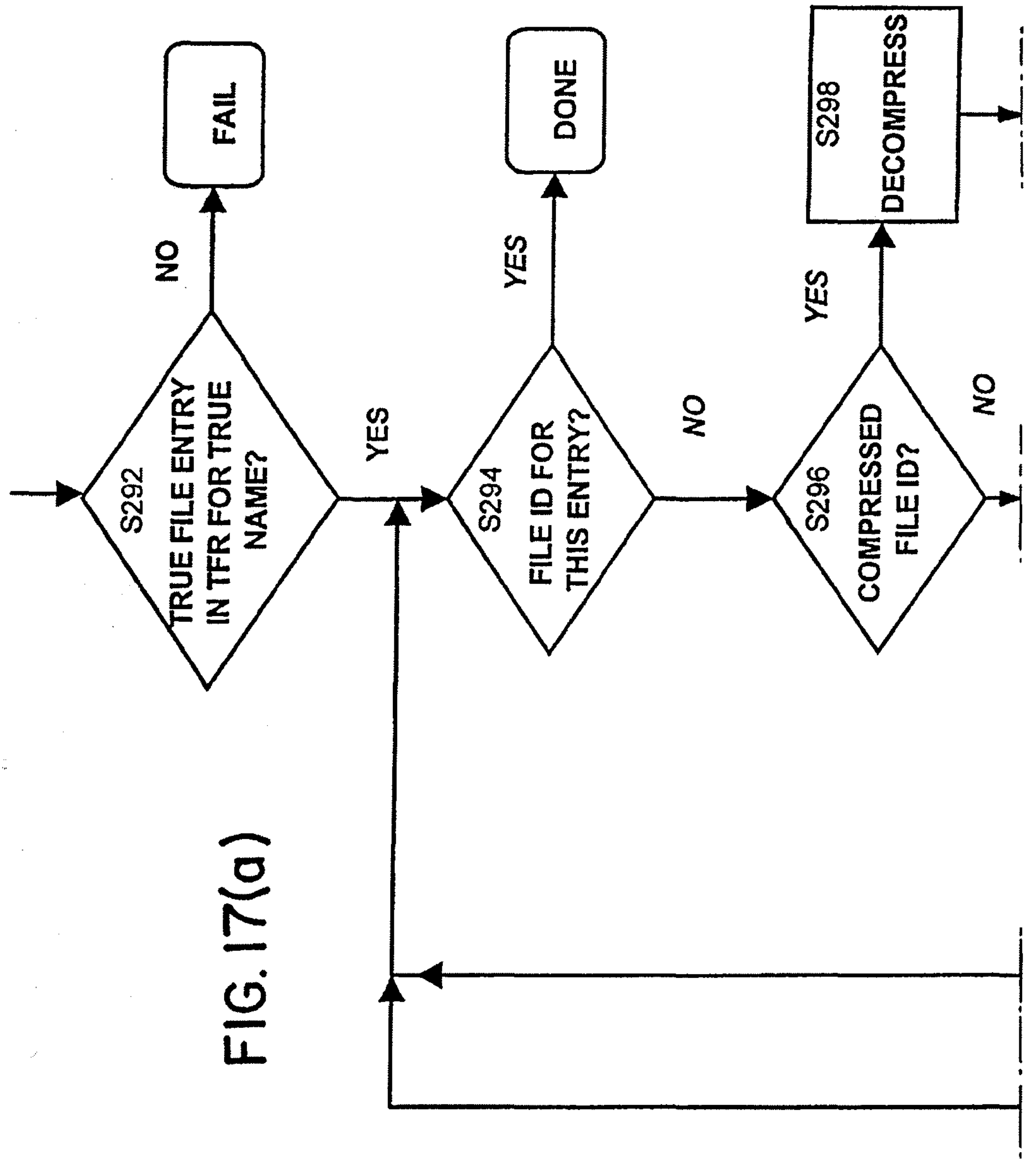

S292
TRUE FILE ENTRY IN TFR FOR TRUE NAME?
NO
FAIL
YES
S294
FILE ID FOR THIS ENTRY?
YES
DONE
NO
S296
COMPRESSED FILE ID?
YES
S298
DECOMPRESS
NO

FIG. 17(a)

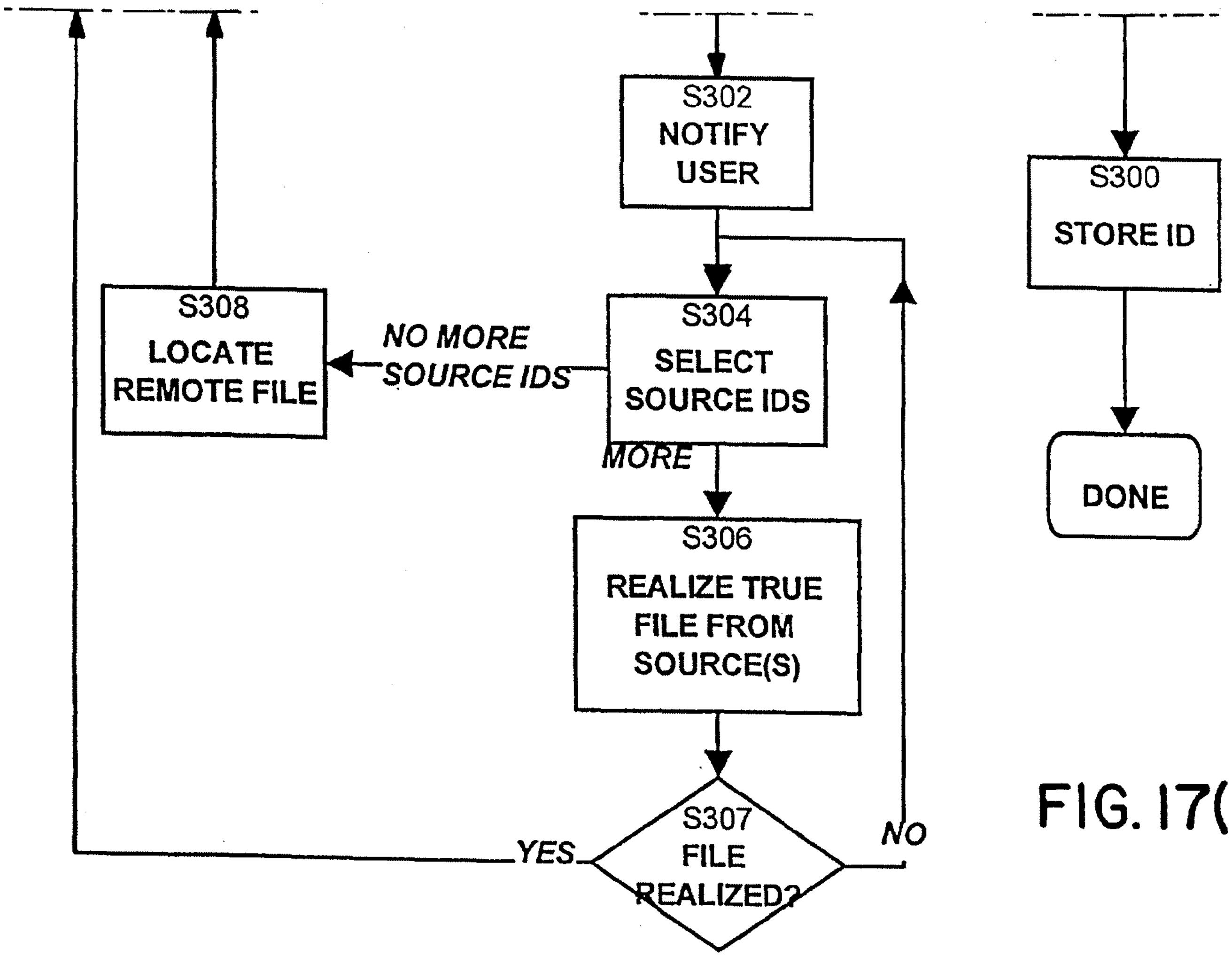

S302
NOTIFY USER
S300
STORE ID
DONE
S308
LOCATE REMOTE FILE
NO MORE SOURCE IDS
S304
SELECT SOURCE IDS
MORE
S306
REALIZE TRUE FILE FROM SOURCE(S)
YES
S307
FILE REALIZED?
NO

FIG. 17(b)

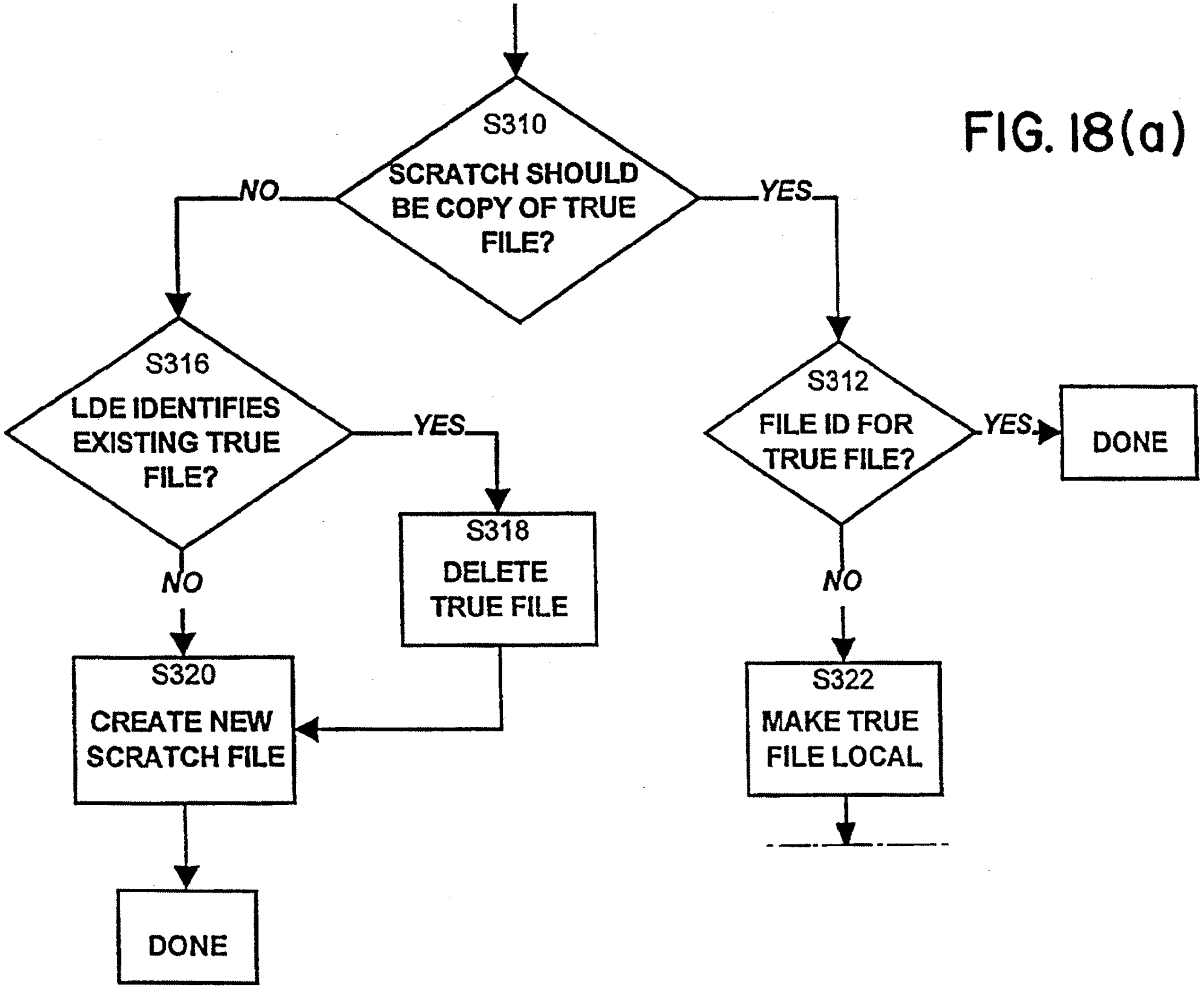

S310
SCRATCH SHOULD BE COPY OF TRUE FILE?
NO
YES
S316
LDE IDENTIFIES EXISTING TRUE FILE?
YES
NO
S318
DELETE TRUE FILE
S320
CREATE NEW SCRATCH FILE
DONE
S312
FILE ID FOR TRUE FILE?
YES
DONE
NO
S322
MAKE TRUE FILE LOCAL

FIG. 18(a)

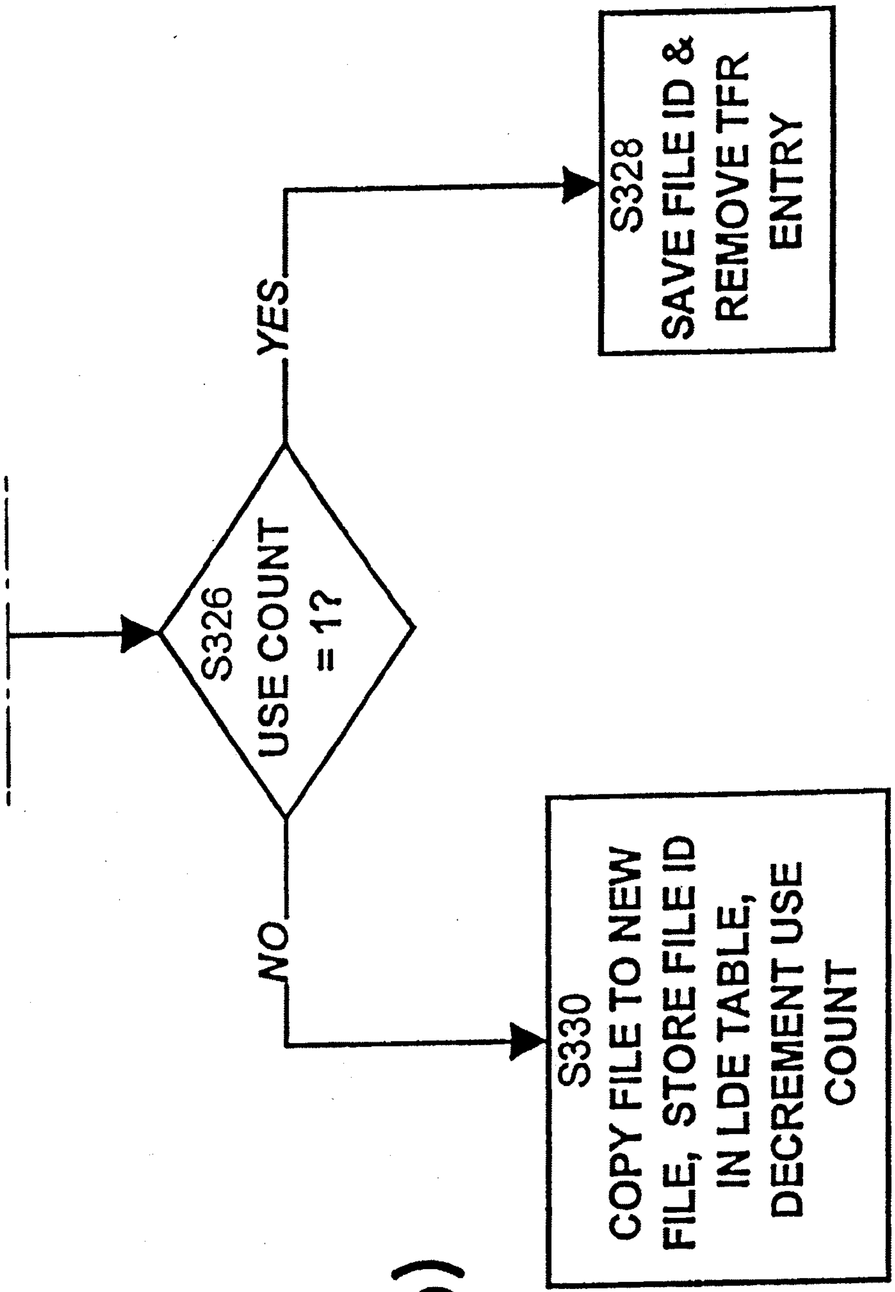

S326
USE COUNT
= 1?
YES
S328
SAVE FILE ID &
REMOVE TFR
ENTRY
NO
S330
COPY FILE TO NEW
FILE, STORE FILE ID
IN LDE TABLE,
DECREMENT USE
COUNT

FIG. 18(b)

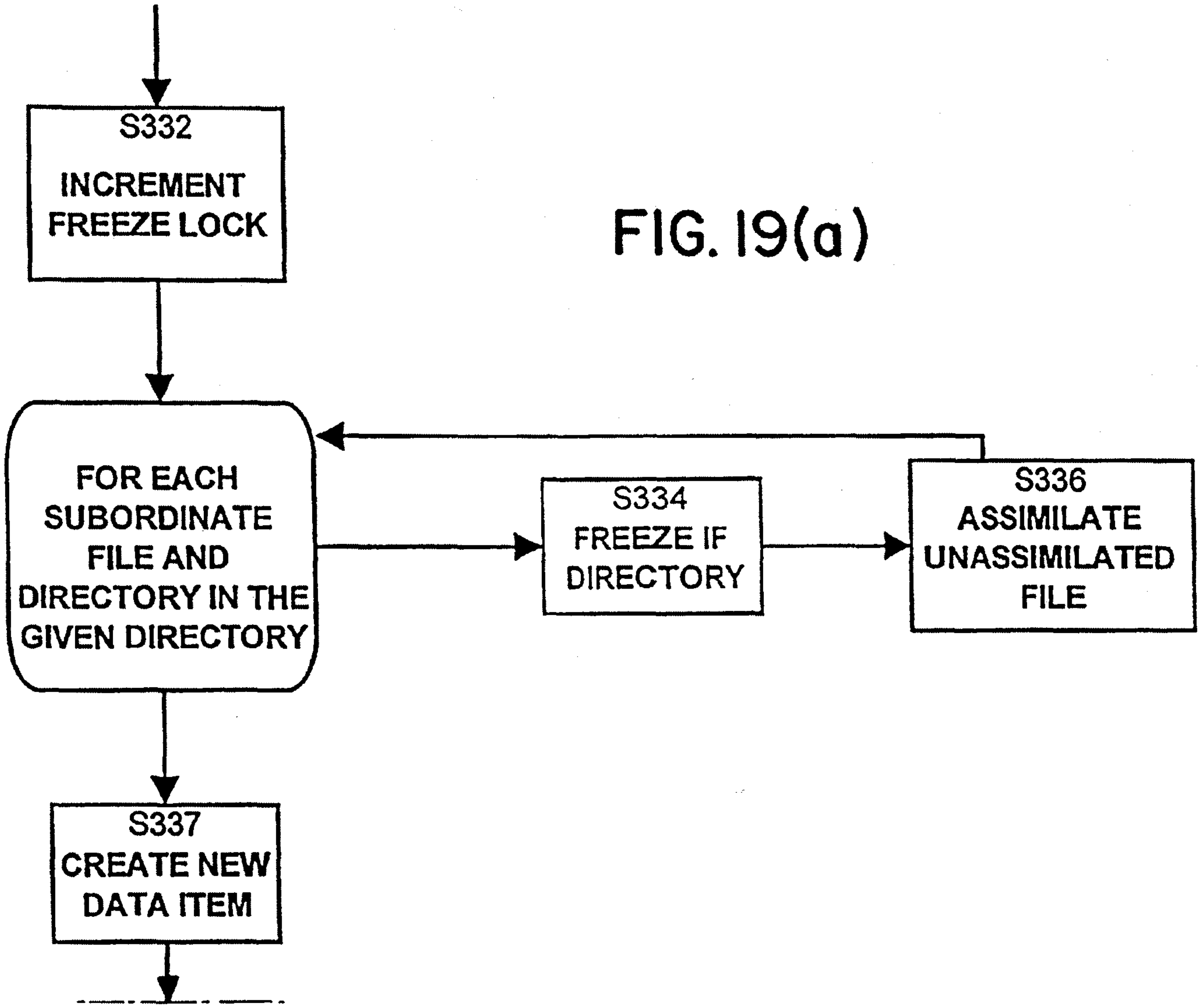

S332
INCREMENT FREEZE LOCK
FOR EACH SUBORDINATE FILE AND DIRECTORY IN THE GIVEN DIRECTORY
S334
FREEZE IF DIRECTORY
S336
ASSIMILATE UNASSIMILATED FILE
S337
CREATE NEW DATA ITEM

FIG. 19(a)

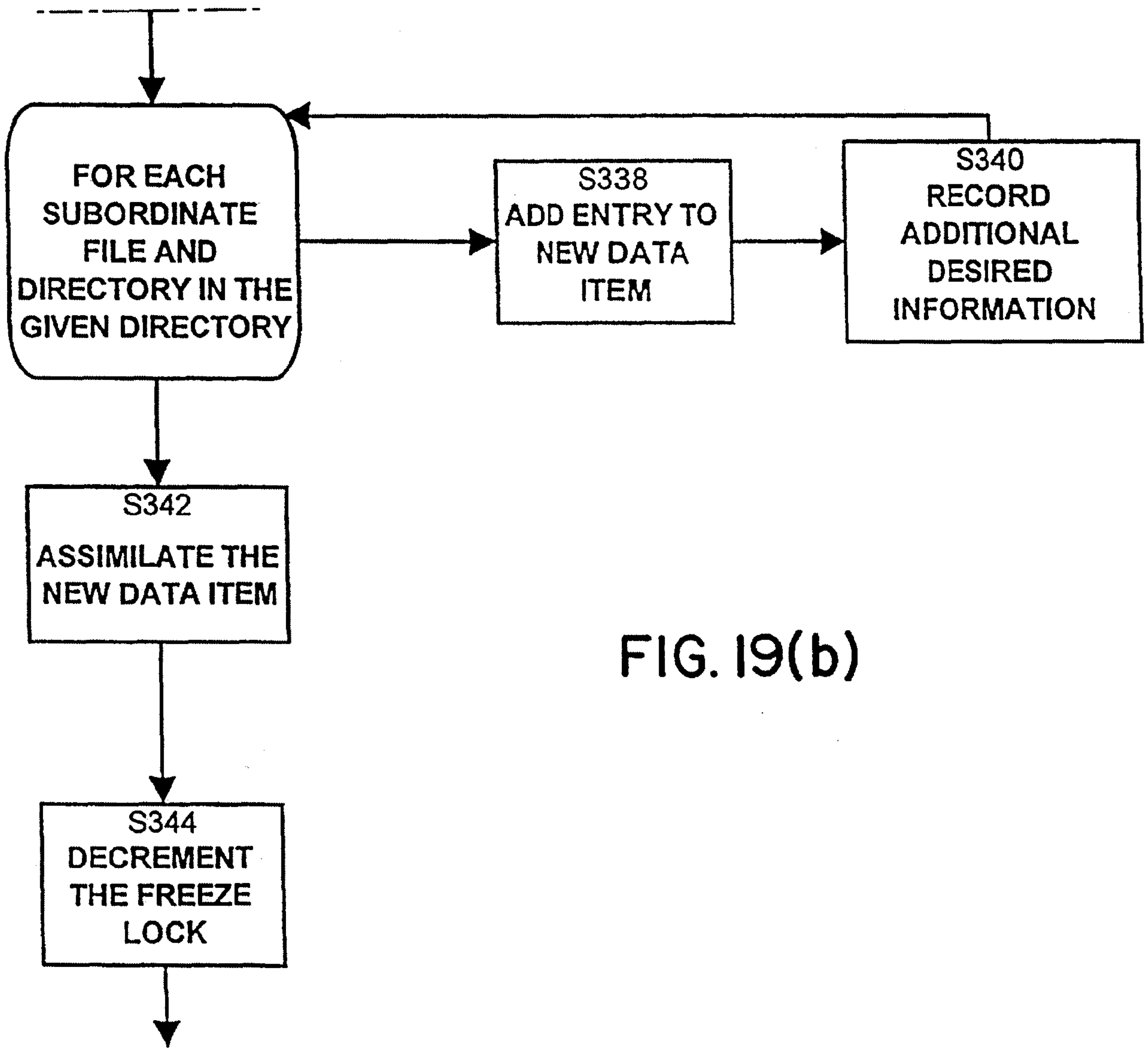
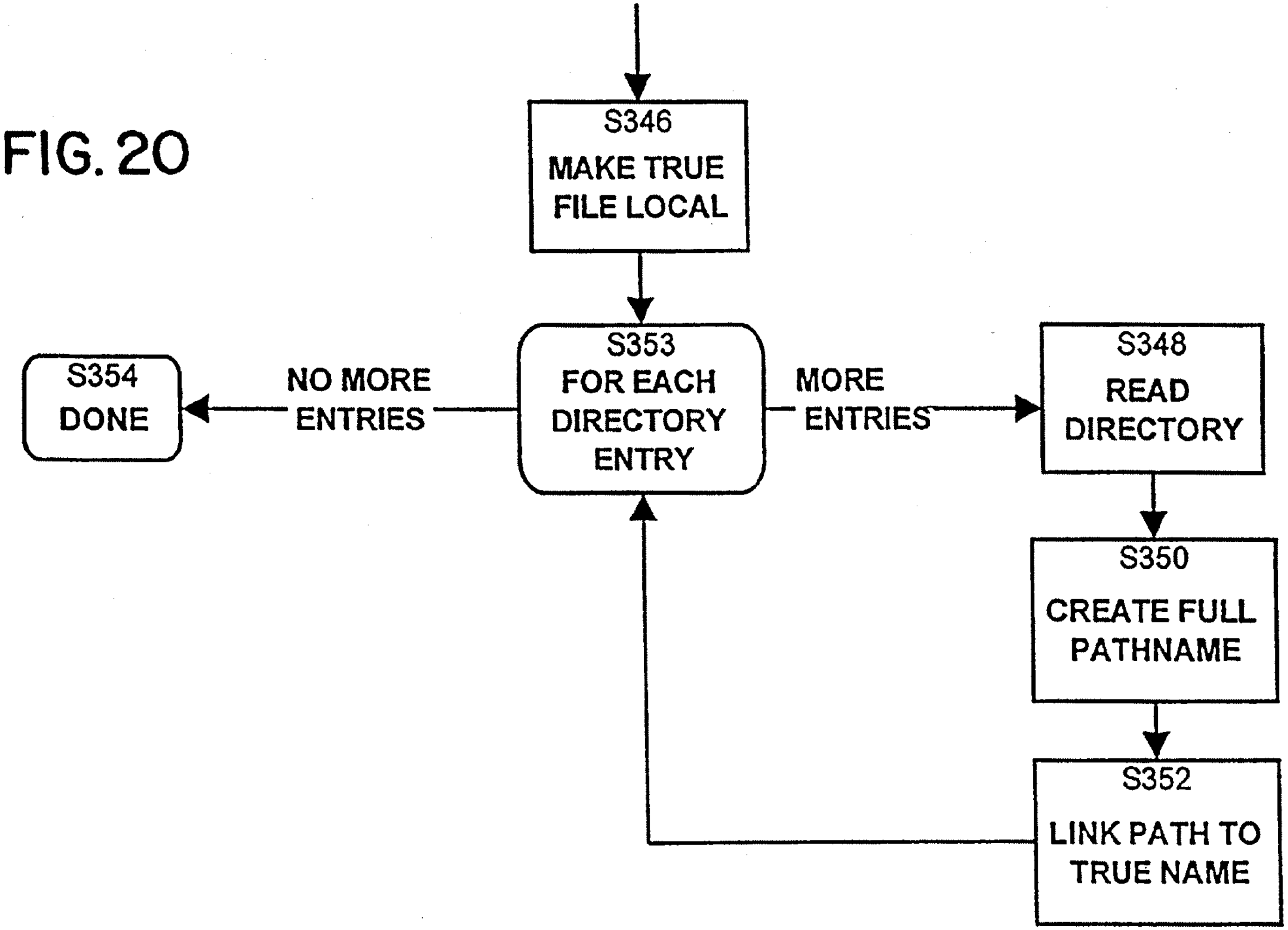

FOR EACH SUBORDINATE FILE AND DIRECTORY IN THE GIVEN DIRECTORY
S338
ADD ENTRY TO NEW DATA ITEM
S340
RECORD ADDITIONAL DESIRED INFORMATION
S342
ASSIMILATE THE NEW DATA ITEM
S344
DECREMENT THE FREEZE LOCK

FIG. 19(b)

FIG. 20


S346
MAKE TRUE FILE LOCAL
S353
FOR EACH DIRECTORY ENTRY
NO MORE ENTRIES
S354
DONE
MORE ENTRIES
S348
READ DIRECTORY
S350
CREATE FULL PATHNAME
S352
LINK PATH TO TRUE NAME

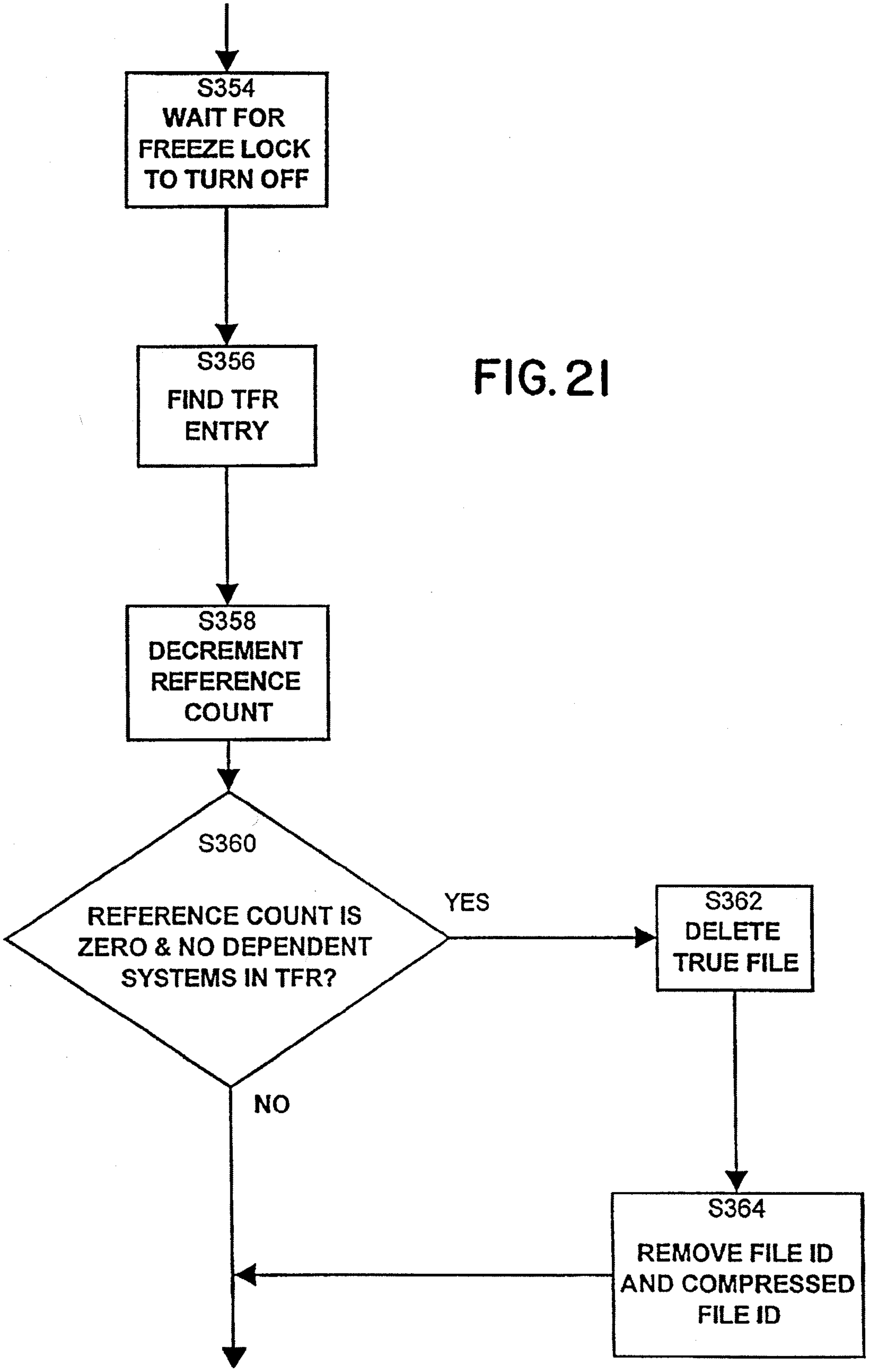

S354
WAIT FOR FREEZE LOCK TO TURN OFF
S356
FIND TFR ENTRY
S358
DECREMENT REFERENCE COUNT
S360
REFERENCE COUNT IS ZERO & NO DEPENDENT SYSTEMS IN TFR?
YES
S362
DELETE TRUE FILE
NO
S364
REMOVE FILE ID AND COMPRESSED FILE ID

FIG. 21

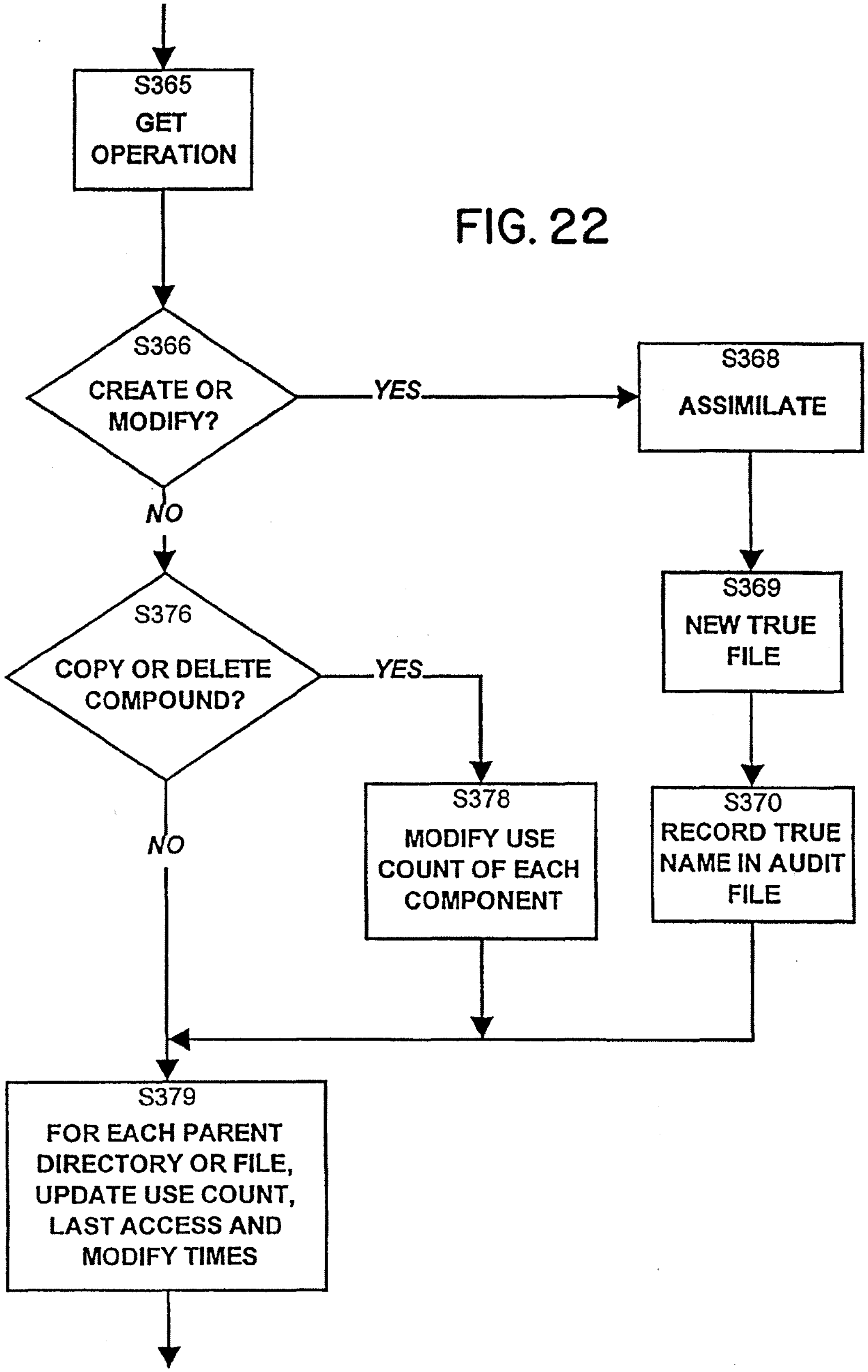
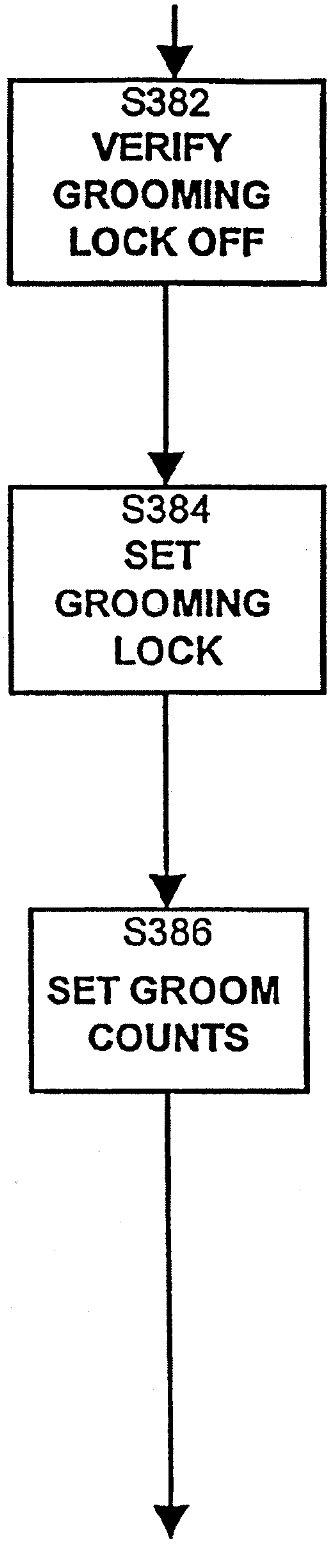

S365
GET OPERATION
S366
CREATE OR MODIFY?
YES
S368
ASSIMILATE
NO
S369
NEW TRUE FILE
S376
COPY OR DELETE COMPOUND?
YES
S378
MODIFY USE COUNT OF EACH COMPONENT
S370
RECORD TRUE NAME IN AUDIT FILE
NO
S379
FOR EACH PARENT DIRECTORY OR FILE, UPDATE USE COUNT, LAST ACCESS AND MODIFY TIMES

FIG. 22

FIG. 23


S382
VERIFY GROOMING LOCK OFF
S384
SET GROOMING LOCK
S386
SET GROOM COUNTS

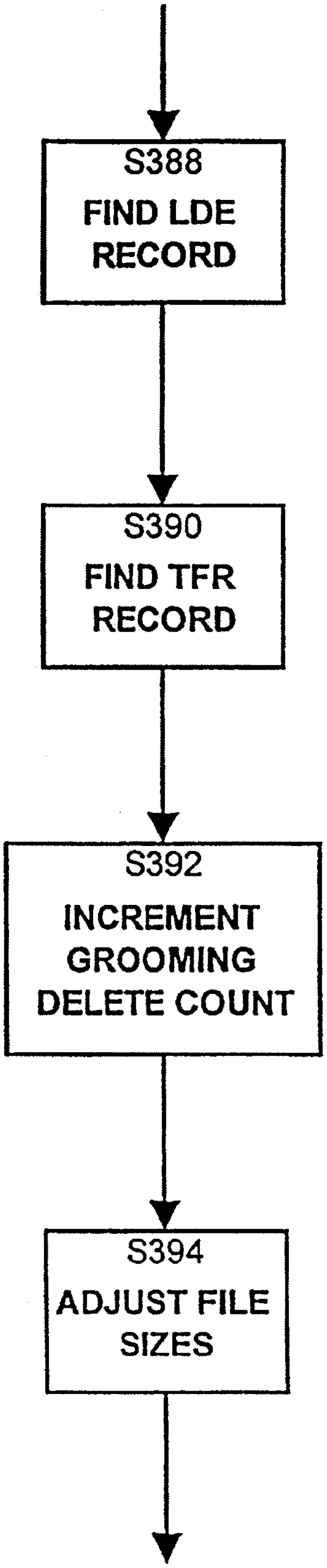

S388
FIND LDE RECORD
S390
FIND TFR RECORD
S392
INCREMENT GROOMING DELETE COUNT
S394
ADJUST FILE SIZES

FIG. 24

FIG. 25

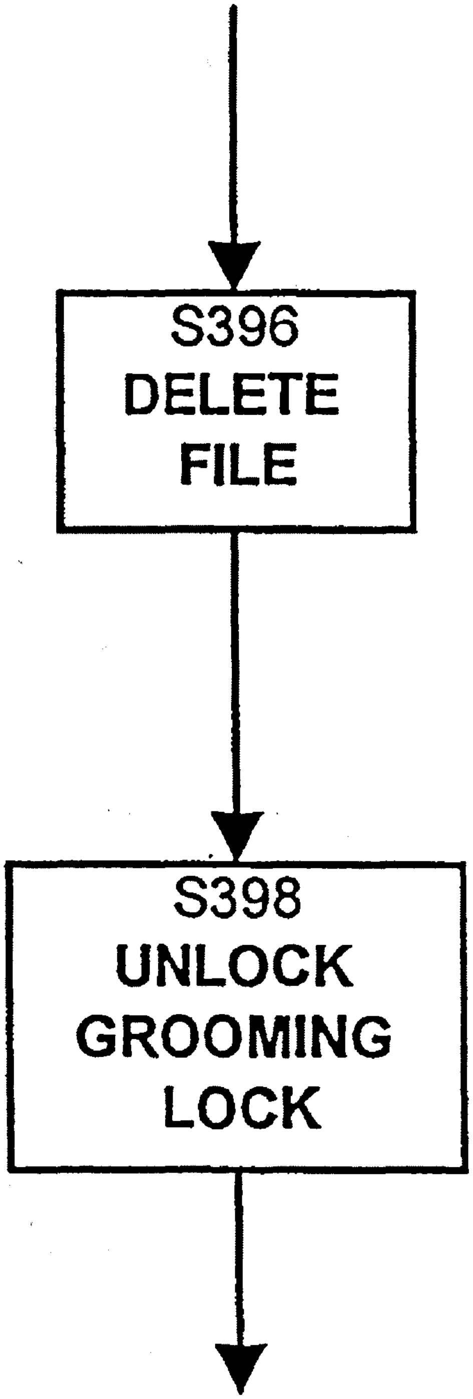
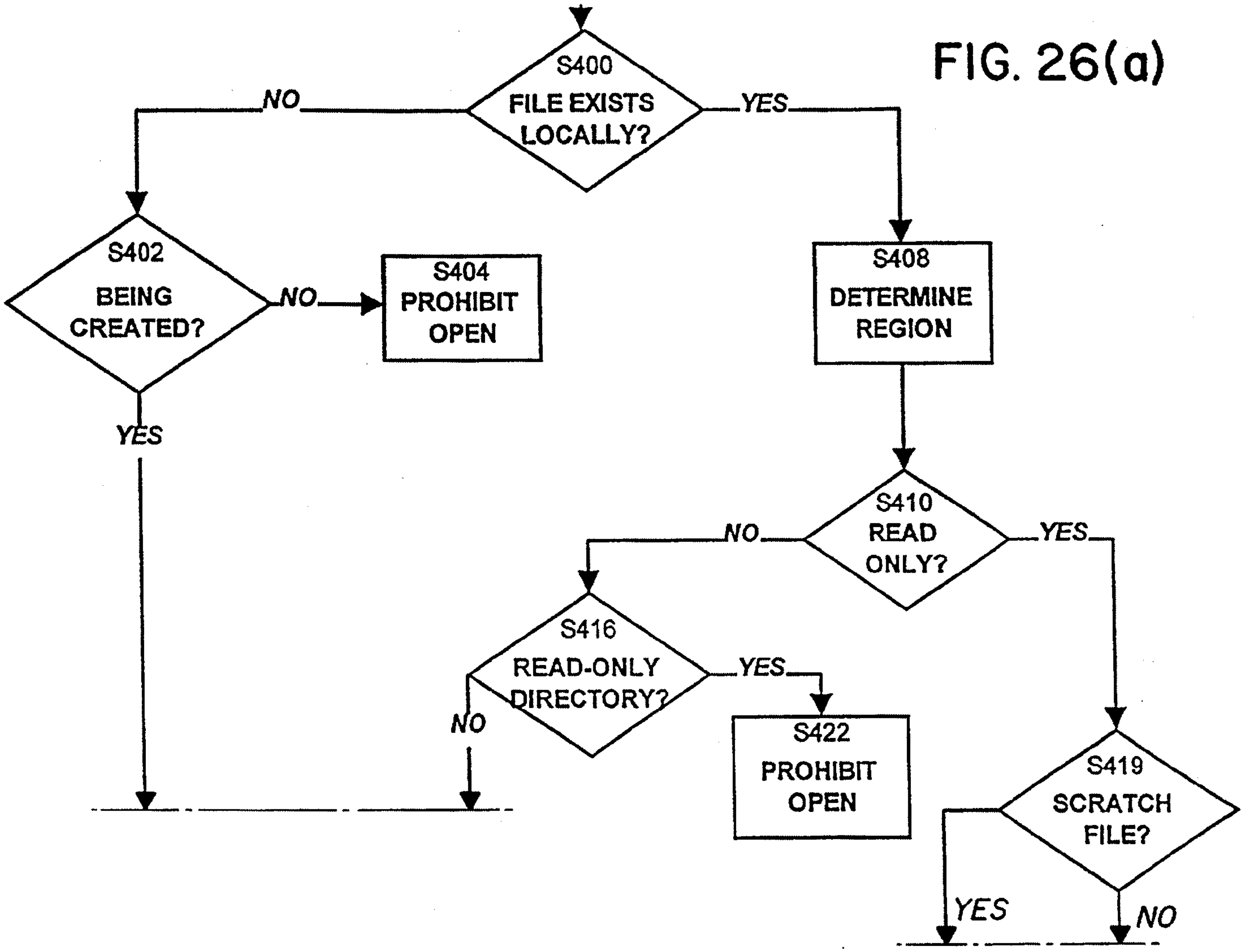


FIG. 26(a)


S400
FILE EXISTS LOCALLY?
NO
YES
S402
BEING CREATED?
NO
S404
PROHIBIT OPEN
YES
S408
DETERMINE REGION
S410
READ ONLY?
NO
YES
S416
READ-ONLY DIRECTORY?
YES
NO
S422
PROHIBIT OPEN
S419
SCRATCH FILE?
YES
NO

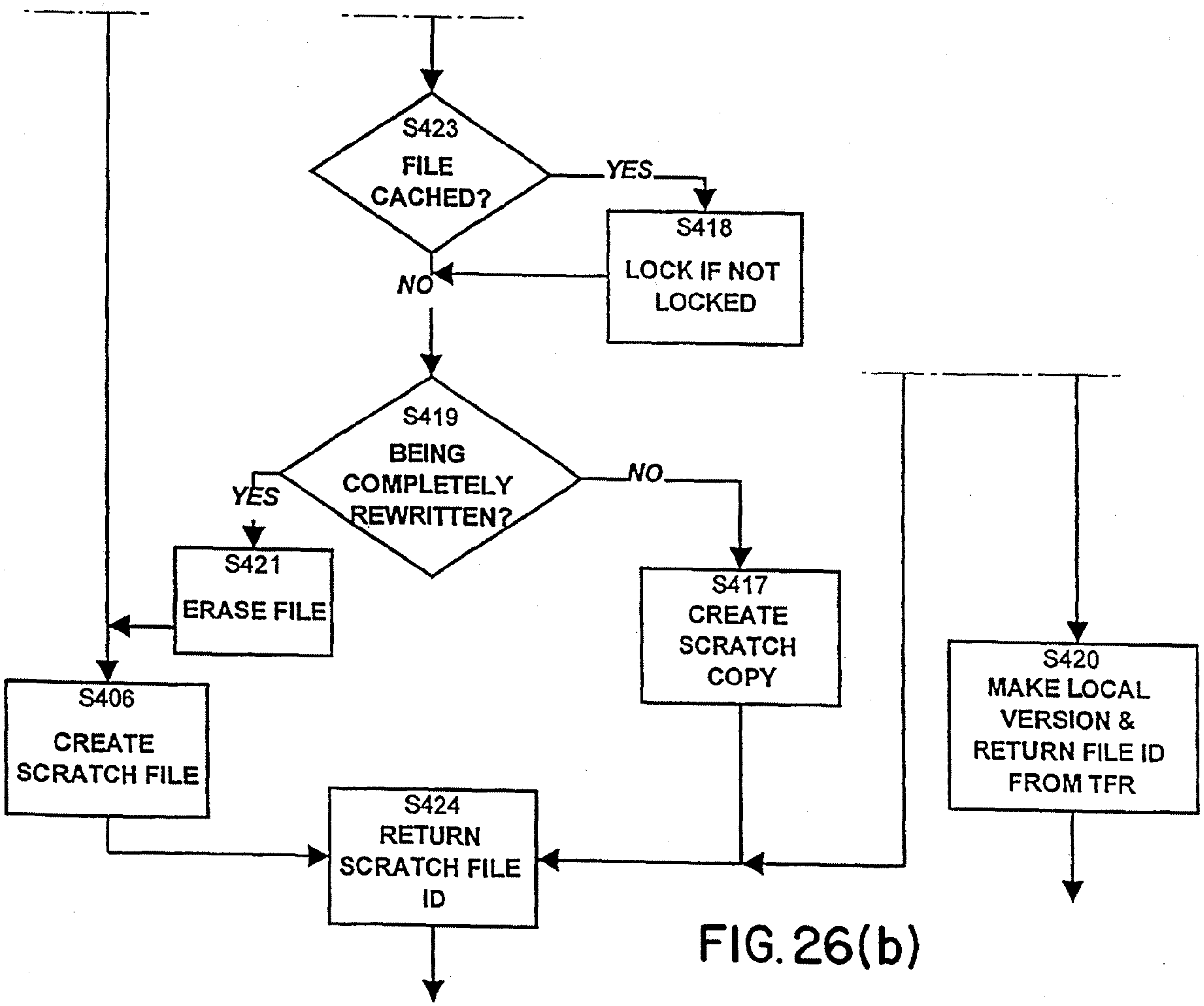

S423
FILE CACHED?
YES
S418
LOCK IF NOT LOCKED
NO
S419
BEING COMPLETELY REWRITTEN?
YES
NO
S421
ERASE FILE
S417
CREATE SCRATCH COPY
S420
MAKE LOCAL VERSION & RETURN FILE ID FROM TFR
S406
CREATE SCRATCH FILE
S424
RETURN SCRATCH FILE ID

FIG. 26(b)

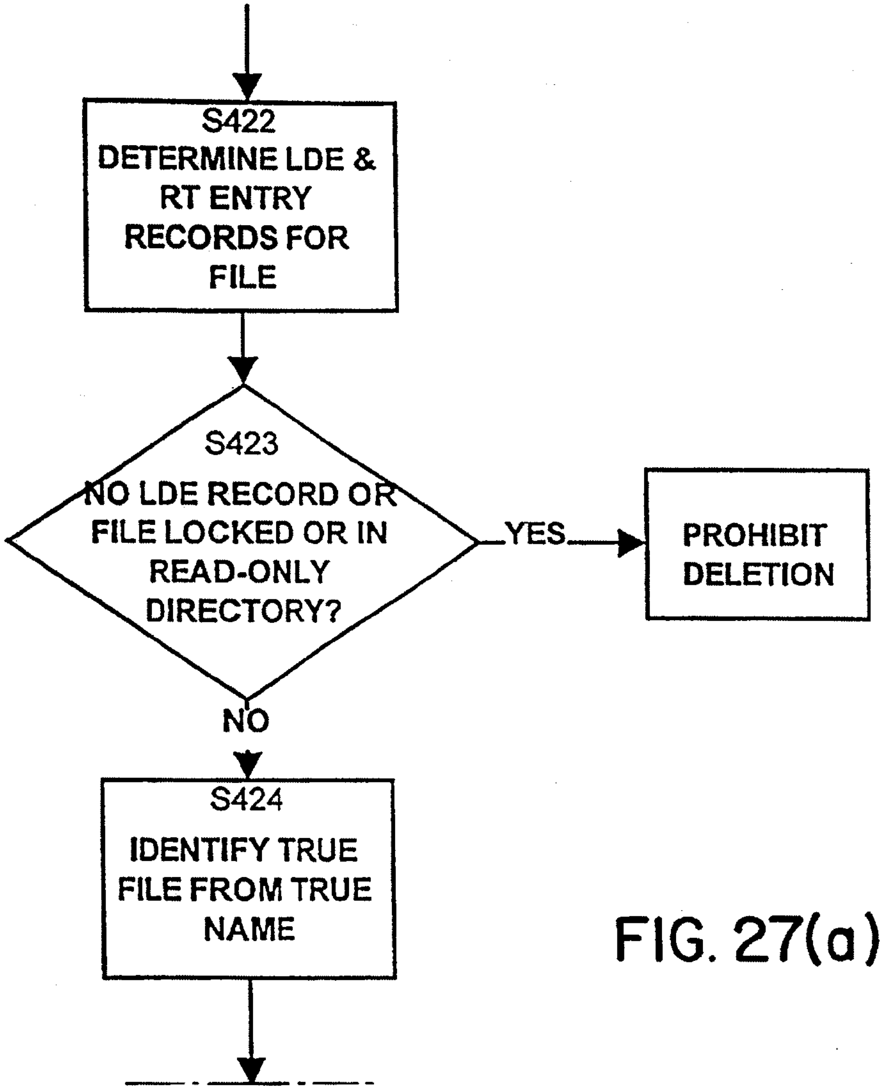

S422
DETERMINE LDE & RT ENTRY RECORDS FOR FILE
S423
NO LDE RECORD OR FILE LOCKED OR IN READ-ONLY DIRECTORY?
YES
PROHIBIT DELETION
NO
S424
IDENTIFY TRUE FILE FROM TRUE NAME

FIG. 27(a)

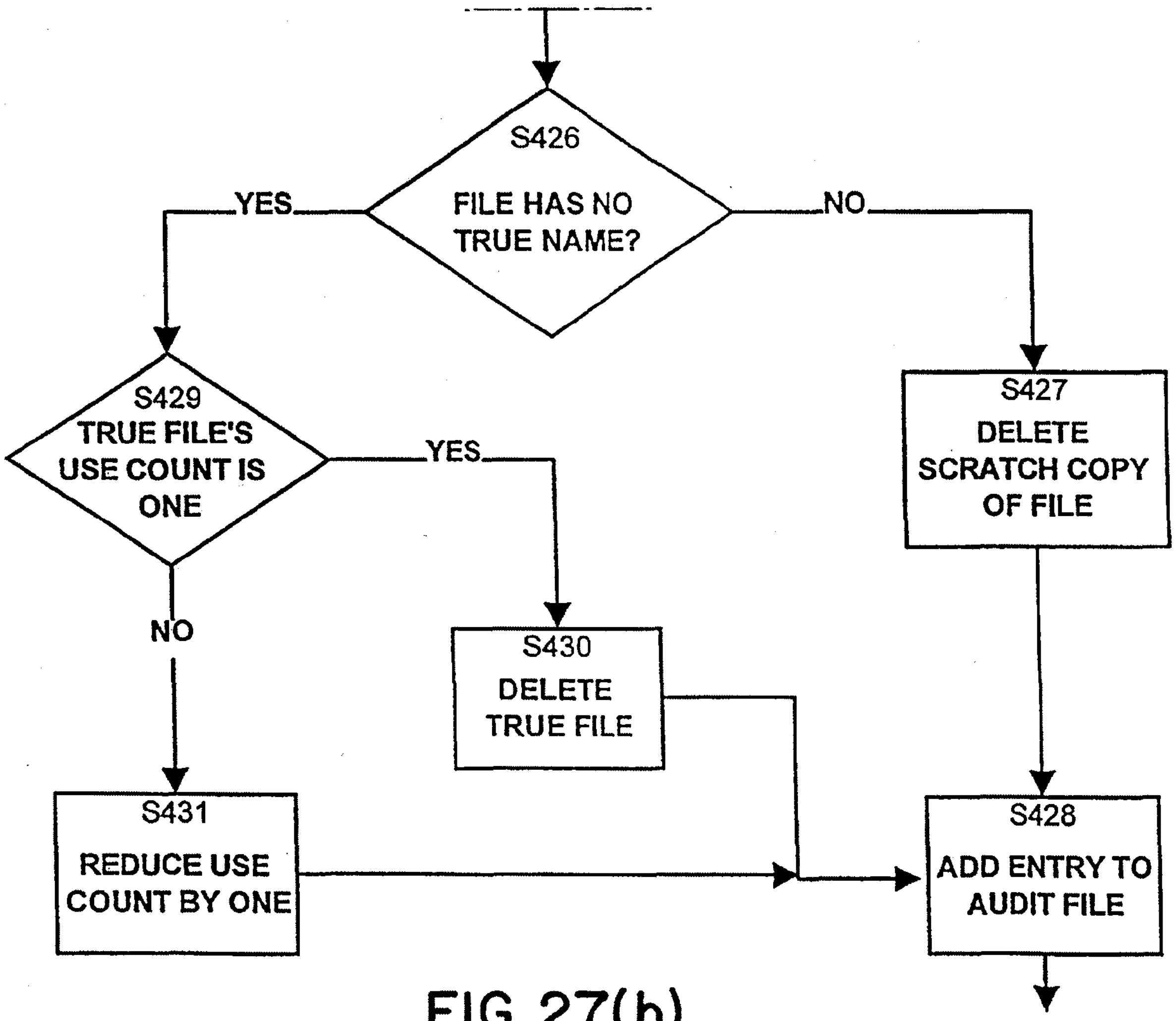
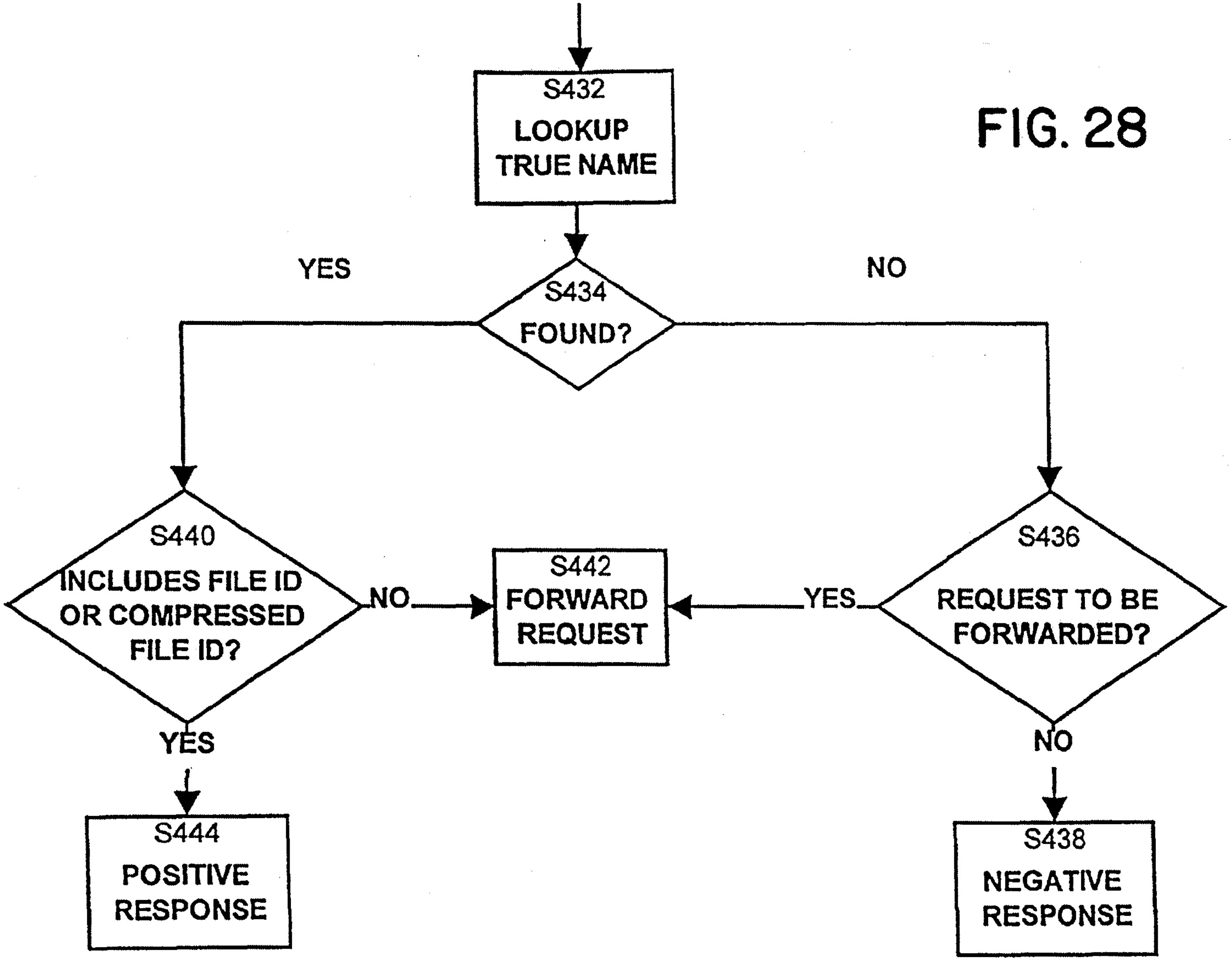

S426
FILE HAS NO TRUE NAME?
YES
NO
S429
TRUE FILE'S USE COUNT IS ONE
YES
NO
S427
DELETE SCRATCH COPY OF FILE
S430
DELETE TRUE FILE
S431
REDUCE USE COUNT BY ONE
S428
ADD ENTRY TO AUDIT FILE

FIG. 27(b)

FIG. 28


S432
LOOKUP TRUE NAME
S434
FOUND?
YES
NO
S440
INCLUDES FILE ID OR COMPRESSED FILE ID?
NO
S442
FORWARD REQUEST
YES
S436
REQUEST TO BE FORWARDED?
YES
NO
S444
POSITIVE RESPONSE
S438
NEGATIVE RESPONSE

DATA PROCESSING SYSTEM USING SUBSTANTIALLY UNIQUE IDENTIFIERS TO IDENTIFY DATA ITEMS, WHEREBY IDENTICAL DATA ITEMS HAVE THE SAME IDENTIFIERS

This is a continuation of application Ser. No. 08/425,160, filed on Apr. 11, 1995, which was abandoned upon the filing hereof.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to data processing systems and, more particularly, to data processing systems wherein data items are identified by substantially unique identifiers which depend on all of the data in the data items and only on the data in the data items.

2. Background of the Invention

Data processing (DP) systems, computers, networks of computers, or the like, typically offer users and programs various ways to identify the data in the systems.

Users typically identify data in the data processing system by giving the data some form of name. For example, a typical operating system (OS) on a computer provides a file system in which data items are named by alphanumeric identifiers. Programs typically identify data in the data processing system using a location or address. For example, a program may identify a record in a file or database by using a record number which serves to locate that record.

In all but the most primitive operating systems, users and programs are able to create and use collections of named data items, these collections themselves being named by identifiers. These named collections can then, themselves, be made part of other named collections. For example, an OS may provide mechanisms to group files (data items) into directories (collections). These directories can then, themselves be made part of other directories. A data item may thus be identified relative to these nested directories using a sequence of names, or a so-called pathname, which defines a path through the directories to a particular data item (file or directory).

As another example, a database management system may group data records (data items) into tables and then group these tables into database files (collections). The complete address of any data record can then be specified using the database file name, the table name, and the record number of that data record.

Other examples of identifying data items include: identifying files in a network file system, identifying objects in an object-oriented database, identifying images in an image database, and identifying articles in a text database.

In general, the terms "data" and "data item" as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits. The term "data processing" herein refers to the processing of data items, and is sometimes dependent on the type of data item being processed. For example, a data processor for a digital image may differ from a data processor for an audio signal.

In all of the prior data processing systems the names or identifiers provided to identify data items (the data items being files, directories, records in the database, objects in

object-oriented programming, locations in memory or on a physical device, or the like) are always defined relative to a specific context. For instance, the file identified by a particular file name can only be determined when the directory containing the file (the context) is known. The file identified by a pathname can be determined only when the file system (context) is known. Similarly, the addresses in a process address space, the keys in a database table, or domain names on a global computer network such as the Internet are meaningful only because they are specified relative to a context.

In prior art systems for identifying data items there is no direct relationship between the data names and the data item. The same data name in two different contexts may refer to different data items, and two different data names in the same context may refer to the same data item.

In addition, because there is no correlation between a data name and the data it refers to, there is no a priori way to confirm that a given data item is in fact the one named by a data name. For instance, in a DP system, if one processor requests that another processor deliver a data item with a given data name, the requesting processor cannot, in general, verify that the data delivered is the correct data (given only the name). Therefore it may require further processing, typically on the part of the requestor, to verify that the data item it has obtained is, in fact, the item it requested.

A common operation in a DP system is adding a new data item to the system. When a new data item is added to the system, a name can be assigned to it only by updating the context in which names are defined. Thus such systems require a centralized mechanism for the management of names. Such a mechanism is required even in a multi-processing system when data items are created and identified at separate processors in distinct locations, and in which there is no other need for communication when data items are added.

In many data processing systems or environments, data items are transferred between different locations in the system. These locations may be processors in the data processing system, storage devices, memory, or the like. For example, one processor may obtain a data item from another processor or from an external storage device, such as a floppy disk, and may incorporate that data item into its system (using the name provided with that data item).

However, when a processor (or some location) obtains a data item from another location in the DP system, it is possible that this obtained data item is already present in the system (either at the location of the processor or at some other location accessible by the processor) and therefore a duplicate of the data item is created. This situation is common in a network data processing environment where proprietary software products are installed from floppy disks onto several processors sharing a common file server. In these systems, it is often the case that the same product will be installed on several systems, so that several copies of each file will reside on the common file server.

In some data processing systems in which several processors are connected in a network, one system is designated as a cache server to maintain master copies of data items, and other systems are designated as cache clients to copy local copies of the master data items into a local cache on an as-needed basis. Before using a cached item, a cache client must either reload the cached item, be informed of changes to the cached item, or confirm that the master item corresponding to the cached item has not changed. In other words, a cache client must synchronize its data items with those on the cache server. This synchronization may involve reloading data items onto the cache client. The need to keep the cache synchronized or reload it adds significant overhead to existing caching mechanisms.

In view of the above and other problems with prior art systems, it is therefore desirable to have a mechanism which allows each processor in a multiprocessor system to determine a common and substantially unique identifier for a data item, using only the data in the data item and not relying on any sort of context.

It is further desirable to have a mechanism for reducing multiple copies of data items in a data processing system and to have a mechanism which enables the identification of identical data items so as to reduce multiple copies. It is further desirable to determine whether two instances of a data item are in fact the same data item, and to perform various other systems' functions and applications on data items without relying on any context information or properties of the data item.

It is also desirable to provide such a mechanism in such a way as to make it transparent to users of the data processing system, and it is desirable that a single mechanism be used to address each of the problems described above.

SUMMARY OF THE INVENTION

This invention provides, in a data processing system, a method and apparatus for identifying a data item in the system, where the identity of the data item depends on all of the data in the data item and only on the data in the data item. Thus the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data, and depends only on the data itself.

This invention further provides an apparatus and a method for determining whether a particular data item is present in the system or at a location in the system, by examining only the data identities of a plurality of data items.

Using the method or apparatus of the present invention, the efficiency and integrity of a data processing system can be improved. The present invention improves the design and operation of a data storage system, file system, relational database, object-oriented database, or the like that stores a plurality of data items, by making possible or improving the design and operation of at least some or all of the following features:

the system stores at most one copy of any data item at a given location, even when multiple data names in the system refer to the same contents;

the system avoids copying data from source to destination locations when the destination locations already have the data;

the system provides transparent access to any data item by reference only to its identity and independent of its present location, whether it be local, remote, or offline;

the system caches data items from a server, so that only the most recently accessed data items need be retained;

when the system is being used to cache data items, problems of maintaining cache consistency are avoided;

the system maintains a desired level of redundancy of data items in a network of servers, to protect against failure by ensuring that multiple copies of the data items are present at different locations in the system;

the system automatically archives data items as they are created or modified;

the system provides the size, age, and location of groups of data items in order to decide whether they can be safely removed from a local file system;

the system can efficiently record and preserve any collection of data items;

the system can efficiently make a copy of any collection of data items, to support a version control mechanism for groups of the data items;

the system can publish data items, allowing other, possibly anonymous, systems in a network to gain access to the data items and to rely on the availability of the data items;

the system can maintain a local inventory of all the data items located on a given removable medium, such as a diskette or CD-ROM, the inventory is independent of other properties of the data items such as their name, location, and date of creation;

the system allows closely related sets of data items, such as matching or corresponding directories on disconnected computers, to be periodically resynchronized with one another;

the system can verify that data retrieved from another location is the desired or requested data, using only the data identifier used to retrieve the data;

the system can prove possession of specific data items by content without disclosing the content of the data items, for purposes of later legal verification and to provide anonymity;

the system tracks possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes.

Other objects, features, and characteristics of the present invention as well as the methods of operation and functions of the related elements of structure, and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. **1**(*a*) and **1**(*b*) depicts a typical data processing system in which a preferred embodiment of the present invention operates;

FIG. **2** depicts a hierarchy of data items stored at any location in such a data processing system;

FIGS. **3**–**9** depict data structures used to implement an embodiment of the present invention; and

FIGS. **10**(*a*)–**28** are flow charts depicting operation of various aspects of the present invention.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EXEMPLARY EMBODIMENTS

An embodiment of the present invention is now described with reference to a typical data processing system **100**, which, with reference to FIGS. **1**(*a*) and **1**(*b*), includes one or more processors (or computers) **102** and various storage devices **104** connected in some way, for example by a bus **106**.

Each processor **102** includes a CPU **108**, a memory **110** and one or more local storage devices **112**. The CPU **108**, memory **110**, and local storage device **112** may be internally connected, for example by a bus **114**. Each processor **102** may also include other devices (not shown), such as a keyboard, a display, a printer, and the like.

In a data processing system **100**, wherein more than one processor **102** is used, that is, in a multiprocessor system, the processors may be in one of various relationships. For example, two processors **102** may be in a client/server, client/client, or a server/server relationship. These inter-processor relationships may be dynamic, changing depending on particular situations and functions. Thus, a particular processor **102** may change its relationship to other processors as needed, essentially setting up a peer-to-peer relationship with other processors. In a peer-to-peer relationship, sometimes a particular processor **102** acts as a client processor, whereas at other times the same processor acts as a server processor. In other words, there is no hierarchy imposed on or required of processors **102**.

In a multiprocessor system, the processors **102** may be homogeneous or heterogeneous. Further, in a multiprocessor data processing system **100**, some or all of the processors **102** may be disconnected from the network of processors for periods of time. Such disconnection may be part of the normal operation of the system **100** or it may be because a particular processor **102** is in need of repair.

Within a data processing system **100**, the data may be organized to form a hierarchy of data storage elements, wherein lower level data storage elements are combined to form higher level elements. This hierarchy can consist of, for example, processors, file systems, regions, directories, data files, segments, and the like. For example, with reference to FIG. **2**, the data items on a particular processor **102** may be organized or structured as a file system **116** which comprises regions **117**, each of which comprises directories **118**, each of which can contain other directories **118** or files **120**. Each file **120** being made up of one or more data segments **122**.

In a typical data processing system, some or all of these elements can be named by users given certain implementation specific naming conventions, the name (or pathname) of an element being relative to a context. In the context of a data processing system **100**, a pathname is fully specified by a processor name, a filesystem name, a sequence of zero or more directory names identifying nested directories, and a final file name. (Usually the lowest level elements, in this case segments **122**, cannot be named by users.)

In other words, a file system **116** is a collection of directories **118**. A directory **118** is a collection of named files **120**—both data files **120** and other directory files **118**. A file **120** is a named data item which is either a data file (which may be simple or compound) or a directory file **118**. A simple file **120** consists of a single data segment **122**. A compound file **120** consists of a sequence of data segments **122**. A data segment **122** is a fixed sequence of bytes. An important property of any data segment is its size, the number of bytes in the sequence.

A single processor **102** may access one or more file systems **116**, and a single storage device **104** may contain one or more file systems **116**, or portions of a file system **116**. For instance, a file system **116** may span several storage devices **104**.

In order to implement controls in a file system, file system **116** may be divided into distinct regions, where each region is a unit of management and control. A region consists of a given directory **118** and is identified by the pathname (user defined) of the directory.

In the following, the term "location", with respect to a data processing system **100**, refers to any of a particular processor **102** in the system, a memory of a particular processor, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any other physical location in the system. The term "local" with respect to a particular processor **102** refers to the memory and storage devices of that particular processor.

In the following, the terms "True Name", "data identity" and "data identifier" refer to the substantially unique data identifier for a particular data item. The term "True File" refers to the actual file, segment, or data item identified by a True Name.

A file system for a data processing system **100** is now described which is intended to work with an existing operating system by augmenting some of the operating system's file management system codes. The embodiment provided relies on the standard file management primitives for actually storing to and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.

The processes and mechanisms (services) provided in this embodiment are grouped into the following categories: primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms.

Primitive mechanisms provide fundamental capabilities used to support other mechanisms. The following primitive mechanisms are described:

1. Calculate True Name;
2. Assimilate Data Item;
3. New True File;
4. Get True Name from Path;
5. Link path to True Name;
6. Realize True File from Location;
7. Locate Remote File;
8. Make True File Local;
9. Create Scratch File;
10. Freeze Directory;
11. Expand Frozen Directory;
12. Delete True File;
13. Process Audit File Entry;
14. Begin Grooming;
15. Select For Removal; and
16. End Grooming.

Operating system mechanisms provide typical familiar file system mechanisms, while maintaining the data structures required to offer the mechanisms of the present invention. Operating system mechanisms are designed to augment existing operating systems, and in this way to make the present invention compatible with, and generally transparent to, existing applications. The following operating system mechanisms are described:

1. Open File;
2. Close File;
3. Read File;
4. Write File;
5. Delete File or Directory;
6. Copy File or Directory;
7. Move File or Directory;
8. Get File Status; and
9. Get Files in Directory.

Remote mechanisms are used by the operating system in responding to requests from other processors. These mechanisms enable the capabilities of the present invention in a peer-to-peer network mode of operation. The following remote mechanisms are described:

1. Locate True File;
2. Reserve True File;
3. Request True File;
4. Retire True File;
5. Cancel Reservation;
6. Acquire True File;
7. Lock Cache;
8. Update Cache; and
9. Check Expiration Date.

Background mechanisms are intended to run occasionally and at a low priority. These provide automated management capabilities with respect to the present invention. The following background mechanisms are described:

1. Mirror True File;
2. Groom Region;
3. Check for Expired Links; and
4. Verify Region; and
5. Groom Source List.

Extended mechanisms run within application programs over the operating system. These mechanisms provide solutions to specific problems and applications. The following extended mechanisms are described:

1. Inventory Existing Directory;
2. Inventory Removable, Read-only Files;
3. Synchronize directories;
4. Publish Region;
5. Retire Directory;
6. Realize Directory at location;
7. Verify True File;
8. Track for accounting purposes; and
9. Track for licensing purposes.

The file system herein described maintains sufficient information to provide a variety of mechanisms not ordinarily offered by an operating system, some of which are listed and described here. Various processing performed by this embodiment of the present invention will now be described in greater detail.

In some embodiments, some files **120** in a data processing system **100** do not have True Names because they have been recently received or created or modified, and thus their True Names have not yet been computed. A file that does not yet have a True Name is called a scratch file. The process of assigning a True Name to a file is referred to as assimilation, and is described later. Note that a scratch file may have a user provided name.

Some of the processing performed by the present invention can take place in a background mode or on a delayed or as-needed basis. This background processing is used to determine information that is not immediately required by the system or which may never be required. As an example, in some cases a scratch file is being changed at a rate greater than the rate at which it is useful to determine its True Name. In these cases, determining the True Name of the file can be postponed or performed in the background.

Data Structures

The following data structures, stored in memory **110** of one of more processors **102** are used to implement the mechanisms described herein. The data structures can be local to each processor **102** of the system **100**, or they can reside on only some of the processors **102**.

The data structures described are assumed to reside on individual peer processors **102** in the data processing system **100**. However, they can also be shared by placing them on a remote, shared file server (for instance, in a local area network of machines). In order to accommodate sharing data structures, it is necessary that the processors accessing the shared database use the appropriate locking techniques to ensure that changes to the shared database do not interfere with one another but are appropriately serialized. These locking techniques are well understood by ordinarily skilled programmers of distributed applications.

It is sometimes desirable to allow some regions to be local to a particular processor **102** and other regions to be shared among processors **102**. (Recall that a region is a unit of file system management and control consisting of a given directory identified by the pathname of the directory.) In the case of local and shared regions, there would be both local and shared versions of each data structure. Simple changes to the processes described below must be made to ensure that appropriate data structures are selected for a given operation.

The local directory extensions (LDE) table **124** is a data structure which provides information about files **120** and directories **118** in the data processing system **100**. The local directory extensions table **124** is indexed by a pathname or contextual name (that is, a user provided name) of a file and includes the True Name for most files. The information in local directory extension table **124** is in addition to that provided by the native file system of the operating system.

The True File registry (TFR) **126** is a data store for listing actual data items which have True Names, both files **120** and segments **122**. When such data items occur in the True File registry **126** they are known as True Files. True Files are identified in True File registry **126** by their True Names or identities. The table True File registry **126** also stores location, dependency, and migration information about True Files.

The region table (RT) **128** defines areas in the network storage which are to be managed separately. Region table **128** defines the rules for access to and migration of files **120** among various regions with the local file system **116** and remote peer file systems.

The source table (ST) **130** is a list of the sources of True Files other than the current True File registry **126**. The source table **130** includes removable volumes and remote processors.

The audit file (AF) **132** is a list of records indicating changes to be made in local or remote files, these changes to be processed in background.

The accounting log (AL) **134** is a log of file transactions used to create accounting information in a manner which preserves the identity of files being tracked independent of their name or location.

The license table (LT) **136** is a table identifying files, which may only be used by licensed users, in a manner independent of their name or location, and the users licensed to use them.

Detailed Descriptions of the Data Structures

The following table summarizes the fields of an local directory extensions table entry, as illustrated by record **138** in FIG. **3**.

Field	Description
Region ID	identifies the region in which this file is contained.
Pathname	the user provided name or contextual name

-continued

Field	Description
	of the file or directory, relative to the region in which it occurs.
True Name	the computed True Name or identity of the file or directory. This True Name is not always up th date, and it is set to a special value when a file is modified and is later recomputed in the background.
Type	indicates whether the file is a data file or a directory.
Scratch File ID	the physical location of the file in the file system, when no True Name has been calculated for the file. As noted above, such a file is called a scratch file.
Time of last access	the last access time to this file. If this file is a directory, this is the last access time to any file in the directory.
Time of last modification	the time of last change of this file. If this file is a directory, this is the last modification time of any file in the directory.
Safe flag	indicates that this file (and, if this file is a directory, all of its subordinate files) have been backed up on some other system, and it is therefore safe to remove them.
Lock flag	indicates whether a file is locked, that is, it is being modified by the local processor or a remote processor. Only one processor may modify a file at a time.
Size	the full size of this directory (including all subordinate files), if all files in it were fully expanded and duplicated. For a file that is not a directory this is the size of the actual True File.
Owner	the identity of the user who owns this file, for accounting and license tracking purposes.

Each record of the True File registry **126** has the fields shown in the True File registry record **140** in FIG. **4**. The True File registry **126** consists of the database described in the table below as well as the actual True Files identified by the True File IDs below.

Field	Description
True Name	computed True Name or identity of the file.
Compressed File ID	compressed version of the True File may be stored insteaded of, or in addition to, an uncompressed version. This field provides the identity of the actual representation of the compressed version of the file.
Grooming delete count	tentative count of how many references have been selected for deletion during a grooming operation.
Time of last access	most recent date and time the content of this file was accessed.
Expiration	date and time after which this file may be deleted by this server.
Dependent processors	processor IDs of other processors which contain references to this True File.
Source IDs	source ID(s) of zero or more sources form which this file or data item may be retrieved.
True File ID	identity or disk location of the actual physical representation of the file or file segment. It is sufficient to use a filename in the registration directory of the underlying operation system. The True File ID is absent if the actual file is not currently present at the current location.
Use count	number of other records on this processor which identify this True File.

A region table **128**, specified by a directory pathname, records storage policies which allow files in the file system to be stored, accessed and migrated in different ways. Storage policies are programmed in a configurable way using a set of rules described below.

Each region table record **142** of region table **128** includes the fields described in the following table (with reference to FIG. **5**):

Field	Description
Region ID	internally used identifier for this region.
Region file system	file system on the local processor of which this region is a part.
Region pathname	a pathname relative to the region file system which defines the location of this region. The region consists of all files and directories subordinate to this pathname, except those in a region subordinate to this region.
Mirror processor(s)	zero or more identifiers of processors which are to keep mirror or archival copies of all files in the current region. Multiple mirror processors can be defined to form a mirror group.
Mirror duplication count	number of copies of each file in this region that should be retained in a mirror group.
Region status	specifies whether this region is local to a single processor 102, shared by several processors 102 (if, for instance, it resides on a shared file server), or managed by a remote processor.
Policy	the migration policy to apply to this region. A single region might participate in several policies. The policies are as follows (parameters in brackets are specified as part of the policy): region is a cached version from [processor ID]; region is a member of a mirror set defined by [processor ID]. region is to be archived on [processor ID]. region is to be backed up locally, by placing new copies in [region ID]. region is read only and may not be changed. region is published and expires on [date]. Files in this region should be compressed.

A source table **130** identifies a source location for True Files. The source table **130** is also used to identify client processors making reservations on the current processor. Each source record **144** of the source table **130** includes the fields summarized in the following table, with reference to FIG. **6**:

Field	Description
source ID	internal identifier used to identify a particular source.
source type	type of source location: Removable Storage Volume Local Region Cache Server Mirror Group Server Cooperative Server Publishing Server Client
source rights	includes information about the rights of this processor, such as whether it can ask the local processor to store data items for it.
source availability	measurement of the bandwidth, cost, and reliability of the connection to this source of True Files. The availability is used to select from among several possible sources.
source location	information on how the local processor is to access the source. This may be, for example, the name of a removable storage volume, or the processor ID and region path of a region on a remote processor.

The audit file **132** is a table of events ordered by timestamp, each record **146** in audit file **132** including the fields summarized in the following table (with reference to FIG. **7**):

Field	Description
Original Name	path of the file in question.
Operation	whether the file was created, read, written, copied or deleted.
Type	specifies whether the source is a file or a directory.
Processor ID	ID of the remote processor generating this event (if not local).
Timestamp	time and date file was closed (required only for accessed/modified files).
Pathname	Name of the file (required only for rename).
True Name	computed True Name of the file. This is used by remote systems to mirror changes to the directory and is filled in during background processing.

Each record **148** of the accounting log **134** records an event which may later be used to provide information for billing mechanisms. Each accounting log entry record **148** includes at least the information summarized in the following table, with reference to FIG. **8**:

Field	Description
date of entry	date and time of this log entry.
type of entry	Entry types include create file, delete file, and transmit file.
True Name	True Name of data item in question.
owner	identity of the user responsible for this action.

Each record **150** of the license table **136** records a relationship between a licensable data item and the user licensed to have access to it. Each license table record **150** includes the information summarized in the following table, with reference to FIG. **9**:

Field	Description
True Name	True Name of a data item subject to license validation.
licensee	identity of a user authorized to have access to this object.

Various other data structures are employed on some or all of the processors **102** in the data processing system **100**. Each processor **102** has a global freeze lock (GFL) **152** (FIG. **1**), which is used to prevent synchronization errors when a directory is frozen or copied. Any processor **102** may include a special archive directory (SAD) **154** into which directories may be copied for the purposes of archival. Any processor **102** may include a special media directory (SMD) **156**, into which the directories of removable volumes are stored to form a media inventory. Each processor has a grooming lock **158**, which is set during a grooming operation. During this period the grooming delete count of True File registry entries **140** is active, and no True Files should be deleted until grooming is complete. While grooming is in effect, grooming information includes a table of pathnames selected for deletion, and keeps track of the amount of space that would be freed if all of the files were deleted.

Primitive Mechanisms

The first of the mechanisms provided by the present invention, primitive mechanisms, are now described. The mechanisms described here depend on underlying data management mechanisms to create, copy, read, and delete data items in the True File registry **126**, as identified by a True File ID. This support may be provided by an underlying operating system or disk storage manager.

The following primitive mechanisms are described:

1. Calculate True Name;
2. Assimilate Data Item;
3. New True File;
4. Get True Name from Path;
5. Link Path to True Name;
6. Realize True File from Location;
7. Locate Remote File;
8. Make True File Local;
9. Create Scratch File;
10. Freeze Directory;
11. Expand Frozen Directory;
12. Delete True File;
13. Process Audit File Entry;
14. Begin Grooming;
15. Select For Removal; and
16. End Grooming.

1. Calculate True Name

A True Name is computed using a function, MD, which reduces a data block B of arbitrary length to a relatively small, fixed size identifier, the True Name of the data block, such that the True Name of the data block is virtually guaranteed to represent the data block B and only data block B.

The function MD must have the following properties:

1. The domain of the function MD is the set of all data items. The range of the function MD is the set of True Names.
2. The function MD must take a data item of arbitrary length and reduce it to an integer value in the range 0 to N−1, where N is the cardinality of the set of True

Names. That is, for an arbitrary length data block B, 0≦MD(B)≦N.

3. The results of MD(B) must be evenly and randomly distributed over the range of N, in such a way that simple or regular changes to B are virtually guaranteed to produce a different value of MD(B).

4. It must be computationally difficult to find a different value B' such that MD(B)=MD(B').

5. The function MD(B) must be efficiently computed.

A family of functions with the above properties are the so-called message digest functions, which are used in digital security systems as techniques for authentification of data. These functions (or algorithms) include MD4, MD5, and SHA.

In the presently preferred embodiments, either MD5 or SHA is employed as the basis for the computation of True Names. Whichever of these two message digest functions is employed, that same function must be employed on a system-wide basis.

It is impossible to define a function having a unique output for each possible input when the number of elements in the range of the function is smaller than the number of elements in its domain. However, a crucial observation is that the actual data items that will be encountered in the operation of any system embodying this invention form a very sparse subset of all the possible inputs.

A colliding set of data items is defined as a set wherein, for one or more pairs x and y in the set, MD(x)=MD(y). Since a function conforming to the requirements for MD must evenly and randomly distribute its outputs, it is possible, by making the range of the function large enough, to make the probability arbitrarily small that actual inputs encountered in the operation of an embodiment of this invention will form a colliding set.

To roughly quantify the probability of a collision, assume that there are no more than 2^{30} storage devices in the world, and that each storage device has an average of at most 2^{20} different data items. Then there are at most 2^{50} data items in the world. If the outputs of MD range between 0 and 2^{128}, it can be demonstrated that the probability of a collision is approximately 1 in 2^{29}. Details on the derivation of these probability values are found, for example, in P. Flajolet and A. M. Odlyzko, "Random Mapping Statistics," *Lecture Notes in Computer Science* 434: *Advances in Cryptology—Eurocrypt '89 Proceedings*, Springer-Verlag, pp. 329–354.

Note that for some less-preferred embodiments of the present invention, lower probabilities of uniqueness may be acceptable, depending on the types of applications and mechanisms used. In some embodiments it may also be useful to have more than one level of True Names, with some of the True Names having different degrees of uniqueness. If such a scheme is implemented, it is necessary to ensure that less unique True Names are not propagated in the system.

While the invention is described herein using only the True Name of a data item as the identifier for the data item, other preferred embodiments use tagged, typed, categorized or classified data items and use a combination of both the True Name and the tag, type, category or class of the data item as an identifier. Examples of such categorizations are files, directories, and segments; executable files and data files, and the like. Examples of classes are classes of objects in an object-oriented system. In such a system, a lower degree of True Name uniqueness is acceptable over the entire universe of data items, as long as sufficient uniqueness is provided per category of data items. This is because the tags provide an additional level of uniqueness.

A mechanism for calculating a True Name given a data item is now described, with reference to FIGS. **10**(*a*) and **10**(*b*).

A simple data item is a data item whose size is less than a particular given size (which must be defined in each particular implementation of the invention). To determine the True Name of a simple data item, with reference to FIG. **10**(*a*), first compute the MD function (described above) on the given simple data item (Step **S212**). Then append to the resulting 128 bits, the byte length modulo **32** of the data item (Step **S214**). The resulting 160-bit value is the True Name of the simple data item.

A compound data item is one whose size is greater than the particular given size of a simple data item. To determine the True Name of an arbitrary (simple or compound) data item, with reference to FIG. **10**(*b*), first determine if the data item is a simple or a compound data item (Step **S216**). If the data item is a simple data item, then compute its True Name in step **S218** (using steps **S212** and **S214** described above), otherwise partition the data item into segments (Step **S220**) and assimilate each segment (Step **S222**) (the primitive mechanism, Assimilate a Data Item, is described below), computing the True Name of the segment. Then create an indirect block consisting of the computed segment True Names (Step **S224**). An indirect block is a data item which consists of the sequence of True Names of the segments. Then, in step **S226**, assimilate the indirect block and compute its True Name. Finally, replace the final thirty-two (32) bits of the resulting True Name (that is, the length of the indirect block) by the length modulo **32** of the compound data item (Step **S228**). The result is the True Name of the compound data item.

Note that the compound data item may be so large that the indirect block of segment True Names is itself a compound data item. In this case the mechanism is invoked recursively until only simple data items are being processed.

Both the use of segments and the attachment of a length to the True Name are not strictly required in a system using the present invention, but are currently considered desirable features in the preferred embodiment.

2. Assimilate Data Item

A mechanism for assimilating a data item (scratch file or segment) into a file system, given the scratch file ID of the data item, is now described with reference to FIG. **11**. The purpose of this mechanism is to add a given data item to the True File registry **126**. If the data item already exists in the True File registry **126**, this will be discovered and used during this process, and the duplicate will be eliminated.

Thereby the system stores at most one copy of any data item or file by content, even when multiple names refer to the same content.

First, determine the True Name of the data item corresponding to the given scratch File ID using the Calculate True Name primitive mechanism (Step **S230**). Next, look for an entry for the True Name in the True File registry **126** (Step **S232**) and determine whether a True Name entry, record **140**, exists in the True File registry **126**. If the entry record includes a corresponding True File ID or compressed File ID (Step **S237**), delete the file with the scratch File ID (Step **S238**). Otherwise store the given True File ID in the entry record (step **S239**).

If it is determined (in step **S232**) that no True Name entry exists in the True File registry **126**, then, in Step **S236**, create a new entry in the True File registry **126** for this True Name. Set the True Name of the entry to the calculated True Name, set the use count for the new entry to one, store the given True File ID in the entry and set the other fields of the entry as appropriate.

Because this procedure may take some time to compute, it is intended to run in background after a file has ceased to change. In the meantime, the file is considered an unassimilated scratch file.

3. New True File

The New True File process is invoked when processing the audit file **132**, some time after a True File has been assimilated (using the Assimilate Data Item primitive mechanism). Given a local directory extensions table entry record **138** in the local directory extensions table **124**, the New True File process can provide the following steps (with reference to FIG. **12**), depending on how the local processor is configured:

First, in step **S238**, examine the local directory extensions table entry record **138** to determine whether the file is locked by a cache server. If the file is locked, then add the ID of the cache server to the dependent processor list of the True File registry table **126**, and then send a message to the cache server to update the cache of the current processor using the Update Cache remote mechanism (Step **242**).

If desired, compress the True File (Step **S246**), and, if desired, mirror the True File using the Mirror True File background mechanism (Step **S248**).

4. Get True Name from Path

The True Name of a file can be used to identify a file by contents, to confirm that a file matches its original contents, or to compare two files. The mechanism to get a True Name given the pathname of a file is now described with reference to FIG. **13**.

First, search the local directory extensions table **124** for the entry record **138** with the given pathname (Step **S250**). If the pathname is not found, this process fails and no True Name corresponding to the given pathname exists. Next, determine whether the local directory extensions table entry record **138** includes a True Name (Step **S252**), and if so, the mechanism's task is complete. Otherwise, determine whether the local directory extensions table entry record **138** identifies a directory (Step **S254**), and if so, freeze the directory (Step **S256**) (the primitive mechanism Freeze Directory is described below).

Otherwise, in step **S258**, assimilate the file (using the Assimilate Data Item primitive mechanism) defined by the File ID field to generate its True Name and store its True Name in the local directory extensions entry record. Then return the True Name identified by the local directory extensions table **124**.

5. Link Path to True Name

The mechanism to link a path to a True Name provides a way of creating a new directory entry record identifying an existing, assimilated file. This basic process may be used to copy, move, and rename files without a need to copy their contents. The mechanism to link a path to a True Name is now described with reference to FIG. **14**.

First, if desired, confirm that the True Name exists locally by searching for it in the True Name registry or local directory extensions table **135** (Step **S260**). Most uses of this mechanism will require this form of validation. Next, search for the path in the local directory extensions table **135** (Step **S262**). Confirm that the directory containing the file named in the path already exists (Step **S264**). If the named file itself exists, delete the File using the Delete True File operating system mechanism (see below) (Step **S268**).

Then, create an entry record in the local directory extensions with the specified path (Step **S270**) and update the entry record and other data structures as follows: fill in the True Name field of the entry with the specified True Name; increment the use count for the True File registry entry record **140** of the corresponding True Name; note whether the entry is a directory by reading the True File to see if it contains a tag (magic number) indicating that it represents a frozen directory (see also the description of the Freeze Directory primitive mechanism regarding the tag); and compute and set the other fields of the local directory extensions appropriately. For instance, search the region table **128** to identify the region of the path, and set the time of last access and time of last modification to the current time.

6. Realize True File from Location

This mechanism is used to try to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File. This mechanism is now described with reference to FIG. **15**.

First, in step **S272**, determine whether the location specified is a processor. If it is determined that the location specified is a processor, then send a Request True File message (using the Request True File remote mechanism) to the remote processor and wait for a response (Step **S274**). If a negative response is received or no response is received after a timeout period, this mechanism fails. If a positive response is received, enter the True File returned in the True File registry **126** (Step **S276**). (If the file received was compressed, enter the True File ID in the compressed File ID field.)

If, on the other hand, it is determined in step **S272** that the location specified is not a processor, then, if necessary, request the user or operator to mount the indicated volume (Step **S278**). Then (Step **S280**) find the indicated file on the given volume and assimilate the file using the Assimilate Data Item primitive mechanism. If the volume does not contain a True File registry **126**, search the media inventory to find the path of the file on the volume. If no such file can be found, this mechanism fails.

At this point, whether or not the location is determined (in step **S272**) to be a processor, if desired, verify the True File (in step **S282**).

7. Locate Remote File

This mechanism allows a processor to locate a file or data item from a remote source of True Files, when a specific source is unknown or unavailable. A client processor system may ask one of several or many sources whether it can supply a data object with a given True Name. The steps to perform this mechanism are as follows (with reference to FIGS. **16**(*a*) and **16**(*b*).

The client processor **102** uses the source table **145** to select one or more source processors (Step **S284**). If no source processor can be found, the mechanism fails. Next, the client processor **102** broadcasts to the selected sources a request to locate the file with the given True Name using the Locate True File remote mechanism (Step **S286**). The request to locate may be augmented by asking to propagate this request to distant servers. The client processor then waits for one or more servers to respond positively (Step **S288**). After all servers respond negatively, or after a timeout period with no positive response, the mechanism repeats selection (Step **S284**) to attempt to identify alternative sources. If any selected source processor responds, its processor ID is the result of this mechanism. Store the processor ID in the source field of the True File registry entry record **140** of the given True Name (Step **S290**).

If the source location of the True Name is a different processor or medium than the destination (Step **S290***a*), perform the following steps:

(i) Look up the True File registry entry record **140** for the corresponding True Name, and add the source location ID to the list of sources for the True Name (Step **S290***b*); and

(ii) If the source is a publishing system, determine the expiration date on the publishing system for the True Name and add that to the list of sources. If the source is not a publishing system, send a message to reserve the True File on the source processor (Step **S290***c*).

Source selection in step **S284** may be based on optimizations involving general availability of the source, access time, bandwidth, and transmission cost, and ignoring previously selected processors which did not respond in step **S288**.

8. Make True File Local

This mechanism is used when a True Name is known and a locally accessible copy of the corresponding file or data item is required. This mechanism makes it possible to actually read the data in a True File. The mechanism takes a True Name and returns when there is a local, accessible copy of the True File in the True File registry **126**. This mechanism is described here with reference to the flow chart of FIGS. **17**(*a*) and **17**(*b*).

First, look in the True File registry **126** for a True File entry record **140** for the corresponding True Name (Step **S292**). If no such entry is found this mechanism fails. If there is already a True File ID for the entry (Step **S294**), this mechanism's task is complete. If there is a compressed file ID for the entry (Step **S296**), decompress the file corresponding to the file ID (Step **S298**) and store the decompressed file ID in the entry (Step **S300**). This mechanism is then complete.

If there is no True File ID for the entry (Step **S294**) and there is no compressed file ID for the entry (Step **S296**), then continue searching for the requested file. At this time it may be necessary to notify the user that the system is searching for the requested file.

If there are one or more source IDs, then select an order in which to attempt to realize the source ID (Step **S304**). The order may be based on optimizations involving general availability of the source, access time, bandwidth, and transmission cost. For each source in the order chosen, realize the True File from the source location (using the Realize True File from Location primitive mechanism), until the True File is realized (Step **S306**). If it is realized, continue with step **S294**. If no known source can realize the True File, use the Locate Remote File primitive mechanism to attempt to find the True File (Step **S308**). If this succeeds, realize the True File from the identified source location and continue with step **S296**.

9. Create Scratch File

A scratch copy of a file is required when a file is being created or is about to be modified. The scratch copy is stored in the file system of the underlying operating system. The scratch copy is eventually assimilated when the audit file record entry **146** is processed by the Process Audit File Entry primitive mechanism. This Create Scratch File mechanism requires a local directory extensions table entry record **138**. When it succeeds, the local directory extensions table entry record **138** contains the scratch file ID of a scratch file that is not contained in the True File registry **126** and that may be modified. This mechanism is now described with reference to FIGS. **18**(*a*) and **18**(*b*).

First determine whether the scratch file should be a copy of the existing True File (Step **S310**). If so, continue with step **S312**. Otherwise, determine whether the local directory extensions table entry record **138** identifies an existing True File (Step **S316**), and if so, delete the True File using the Delete True File primitive mechanism (Step **S318**). Then create a new, empty scratch file and store its scratch file ID in the local directory extensions table entry record **138** (Step **S320**). This mechanism is then complete.

If the local directory extensions table entry record **138** identifies a scratch file ID (Step **S312**), then the entry already has a scratch file, so this mechanism succeeds.

If the local directory extensions table entry record **138** identifies a True File (**S316**), and there is no True File ID for the True File (**S312**), then make the True File local using the Make True File Local primitive mechanism (Step **S322**). If there is still no True File ID, this mechanism fails.

There is now a local True File for this file. If the use count in the corresponding True File registry entry record **140** is one (Step **S326**), save the True File ID in the scratch file ID of the local directory extensions table entry record **138**, and remove the True File registry entry record **140** (Step **S328**). (This step makes the True File into a scratch file.) This mechanism's task is complete.

Otherwise, if the use count in the corresponding True File registry entry record **140** is not one (in step **S326**), copy the file with the given True File ID to a new scratch file, using the Read File OS mechanism and store its file ID in the local directory extensions table entry record **138** (Step **S330**), and reduce the use count for the True File by one. If there is insufficient space to make a copy, this mechanism fails.

10. Freeze Directory

This mechanism freezes a directory in order to calculate its True Name. Since the True Name of a directory is a function of the files within the directory, they must not change during the computation of the True Name of the directory. This mechanism requires the pathname of a directory to freeze. This mechanism is described with reference to FIGS. **19**(*a*) and **19**(*b*).

In step **S332**, add one to the global freeze lock. Then search the local directory extensions table **124** to find each subordinate data file and directory of the given directory, and freeze each subordinate directory found using the Freeze Directory primitive mechanism (Step **S334**). Assimilate each unassimilated data file in the directory using the Assimilate Data Item primitive mechanism (Step **S336**). Then create a data item which begins with a tag or marker (a "magic number") being a unique data item indicating that this data item is a frozen directory (Step **S337**). Then list the file name and True Name for each file in the current directory (Step **S338**). Record any additional information required, such as the type, time of last access and modification, and size (Step **S340**). Next, in step **S342**, using the Assimilate Data Item primitive mechanism, assimilate the data item created in step **S338**. The resulting True Name is the True Name of the frozen directory. Finally, subtract one from the global freeze lock (Step **S344**).

11. Expand Frozen Directory

This mechanism expands a frozen directory in a given location. It requires a given pathname into which to expand the directory, and the True Name of the directory and is described with reference to FIG. **20**.

First, in step **S346**, make the True File with the given True Name local using the Make True File Local primitive mechanism. Then read each directory entry in the local file created in step **S346** (Step **S348**). For each such directory entry, do the following:

Create a full pathname using the given pathname and the file name of the entry (Step **S350**); and

link the created path to the True Name (Step **S352**) using the Link Path to True Name primitive mechanism.

12. Delete True File

This mechanism deletes a reference to a True Name. The underlying True File is not removed from the True File registry **126** unless there are no additional references to the file. With reference to FIG. **21**, this mechanism is performed as follows:

If the global freeze lock is on, wait until the global freeze lock is turned off (Step S354). This prevents deleting a True File while a directory which might refer to it is being frozen. Next, find the True File registry entry record **140** given the True Name (Step S356). If the reference count field of the True File registry **126** is greater than zero, subtract one from the reference count field (Step S358). If it is determined (in step S360) that the reference count field of the True File registry entry record **140** is zero, and if there are no dependent systems listed in the True File registry entry record **140**, then perform the following steps:

(i) If the True File is a simple data item, then delete the True File, otherwise,

(ii) (the True File is a compound data item) for each True Name in the data item, recursively delete the True File corresponding to the True Name (Step S362).

(iii) Remove the file indicated by the True File ID and compressed file ID from the True File registry **126**, and remove the True File registry entry record **140** (Step S364).

13. Process Audit File Entry

This mechanism performs tasks which are required to maintain information in the local directory extensions table **124** and True File registry **126**, but which can be delayed while the processor is busy doing more time-critical tasks. Entries **142** in the audit file **132** should be processed at a background priority as long as there are entries to be processed. With reference to FIG. **22**, the steps for processing an entry are as follows:

Determine the operation in the entry **142** currently being processed (Step S365). If the operation indicates that a file was created or written (Step S366), then assimilate the file using the Assimilate Data Item primitive mechanism (Step S368), use the New True File primitive mechanism to do additional desired processing (such as cache update, compression, and mirroring) (Step S369), and record the newly computed True Name for the file in the audit file record entry (Step S370).

Otherwise, if the entry being processed indicates that a compound data item or directory was copied (or deleted) (Step S376), then for each component True Name in the compound data item or directory, add (or subtract) one to the use count of the True File registry entry record **140** corresponding to the component True Name (Step S378).

In all cases, for each parent directory of the given file, update the size, time of last access, and time of last modification, according to the operation in the audit record (Step S379).

Note that the audit record is not removed after processing, but is retained for some reasonable period so that it may be used by the Synchronize Directory extended mechanism to allow a disconnected remote processor to update its representation of the local system.

14. Begin Grooming

This mechanism makes it possible to select a set of files for removal and determine the overall amount of space to be recovered. With reference to FIG. **23**, first verify that the global grooming lock is currently unlocked (Step S382). Then set the global grooming lock, set the total amount of space freed during grooming to zero and empty the list of files selected for deletion (Step S384). For each True File in the True File registry **126**, set the delete count to zero (Step S386).

15. Select For Removal

This grooming mechanism tentatively selects a pathname to allow its corresponding True File to be removed. With reference to FIG. **24**, first find the local directory extensions table entry record **138** corresponding to the given pathname (Step S388). Then find the True File registry entry record **140** corresponding to the True File name in the local directory extensions table entry record **138** (Step S390). Add one to the grooming delete count in the True File registry entry record **140** and add the pathname to a list of files selected for deletion (Step S392). If the grooming delete count of the True File registry entry record **140** is equal to the use count of the True File registry entry record **140**, and if the there are no entries in the dependency list of the True File registry entry record **140**, then add the size of the file indicated by the True File ID and or compressed file ID to the total amount of space freed during grooming (Step S394).

16. End Grooming

This grooming mechanism ends the grooming phase and removes all files selected for removal. With reference to FIG. **25**, for each file in the list of files selected for deletion, delete the file (Step S396) and then unlock the global grooming lock (Step S398).

Operating System Mechanisms

The next of the mechanisms provided by the present invention, operating system mechanisms, are now described.

The following operating system mechanisms are described:

1. Open File;
2. Close File;
3. Read File;
4. Write File;
5. Delete File or Directory;
6. Copy File or Directory;
7. Move File or Directory;
8. Get File Status; and
9. Get Files in Directory.

1. open File

A mechanism to open a file is described with reference to FIGS. **26**(*a*) and **26**(*b*). This mechanism is given as input a pathname and the type of access required for the file (for example, read, write, read/write, create, etc.) and produces either the File ID of the file to be opened or an indication that no file should be opened. The local directory extensions table record **138** and region table record **142** associated with the opened file are associated with the open file for later use in other processing functions which refer to the file, such as read, write, and close.

First, determine whether or not the named file exists locally by examining the local directory extensions table **124** to determine whether there is an entry corresponding to the given pathname (Step S400). If it is determined that the file name does not exist locally, then, using the access type, determine whether or not the file is being created by this opening process (Step S402). If the file is not being created, prohibit the open (Step S404). If the file is being created, create a zero-length scratch file using an entry in local directory extensions table **124** and produce the scratch file ID of this scratch file as the result (Step S406).

If, on the other hand, it is determined in step S**400** that the file name does exist locally, then determine the region in which the file is located by searching the region table **128** to find the record **142** with the longest region path which is a prefix of the file pathname (Step S**408**). This record identifies the region of the specified file.

Next, determine using the access type, whether the file is being opened for writing or whether it is being opened only for reading (Step S410). If the file is being opened for reading only, then, if the file is a scratch file (Step S419), return the scratch File ID of the file (Step S424). Otherwise get the True Name from the local directory extensions table **124** and make a local version of the True File associated with the True Name using the Make True File Local primitive mechanism, and then return the True File ID associated with the True Name (Step **S420**).

If the file is not being opened for reading only (Step **S410**), then, if it is determined by inspecting the region table entry record **142** that the file is in a read-only directory (Step **S416**), then prohibit the opening (Step **S422**).

If it is determined by inspecting the region table **128** that the file is in a cached region (Step **S423**), then send a Lock Cache message to the corresponding cache server, and wait for a return message (Step **S418**). If the return message says the file is already locked, prohibit the opening.

If the access type indicates that the file being modified is being rewritten completely (Step **S419**), so that the original data will not be required, then Delete the File using the Delete File OS mechanism (Step **S421**) and perform step **S406**. Otherwise, make a scratch copy of the file (Step **S417**) and produce the scratch file ID of the scratch file as the result (Step **S424**).

2. Close File

This mechanism takes as input the local directory extensions table entry record **138** of an open file and the data maintained for the open file. To close a file, add an entry to the audit file indicating the time and operation (create, read or write). The audit file processing (using the Process Audit File Entry primitive mechanism) will take care of assimilating the file and thereby updating the other records.

3. Read File

To read a file, a program must provide the offset and length of the data to be read, and the location of a buffer into which to copy the data read.

The file to be read from is identified by an open file descriptor which includes a File ID as computed by the Open File operating system mechanism defined above. The File ID may identify either a scratch file or a True File (or True File segment). If the File ID identifies a True File, it may be either a simple or a compound True File. Reading a file is accomplished by the following steps:

In the case where the File ID identifies a scratch file or a simple True File, use the read capabilities of the underlying operating system.

In the case where the File ID identifies a compound file, break the read operation into one or more read operations on component segments as follows:

A. Identify the segment(s) to be read by dividing the specified file offset and length each by the fixed size of a segment (a system dependent parameter), to determine the segment number and number of segments that must be read.

B. For each segment number computed above, do the following:

i. Read the compound True File index block to determine the True Name of the segment to be read.

ii. Use the Realize True File from Location primitive mechanism to make the True File segment available locally. (If that mechanism fails, the Read File mechanism fails).

iii. Determine the File ID of the True File specified by the True Name corresponding to this segment.

iv. Use the Read File mechanism (recursively) to read from this segment into the corresponding location in the specified buffer.

4. Write File

File writing uses the file ID and data management capabilities of the underlying operating system. File access (Make File Local described above) can be deferred until the first read or write.

5. Delete File or Directory

The process of deleting a file, for a given pathname, is described here with reference to FIGS. **27**(*a*) and **27**(*b*).

First, determine the local directory extensions table entry record **138** and region table entry record **142** for the file (Step **S422**). If the file has no local directory extensions table entry record **138** or is locked or is in a read-only region, prohibit the deletion.

Identify the corresponding True File given the True Name of the file being deleted using the True File registry **126** (Step **S424**). If the file has no True Name, (Step **S426**) then delete the scratch copy of the file based on its scratch file ID in the local directory extensions table **124** (Step **S427**), and continue with step **S428**.

If the file has a True Name and the True File's use count is one (Step **S429**), then delete the True File (Step **S430**), and continue with step **S428**.

If the file has a True Name and the True File's use count is greater than one, reduce its use count by one (Step **S431**). Then proceed with step **S428**.

In Step **S428**, delete the local directory extensions table entry record, and add an entry to the audit file **132** indicating the time and the operation performed (delete).

6. Copy File or Directory

A mechanism is provided to copy a file or directory given a source and destination processor and pathname. The Copy File mechanism does not actually copy the data in the file, only the True Name of the file. This mechanism is performed as follows:

(A) Given the source path, get the True Name from the path. If this step fails, the mechanism fails.

(B) Given the True Name and the destination path, link the destination path to the True Name.

(C) If the source and destination processors have different True File registries, find (or, if necessary, create) an entry for the True Name in the True File registry table **126** of the destination processor. Enter into the source ID field of this new entry the source processor identity.

(D) Add an entry to the audit file **132** indicating the time and operation performed (copy).

This mechanism addresses capability of the system to avoid copying data from a source location to a destination location when the destination already has the data. In addition, because of the ability to freeze a directory, this mechanism also addresses capability of the system immediately to make a copy of any collection of files, thereby to support an efficient version control mechanisms for groups of files.

7. Move File or Directory

A mechanism is described which moves (or renames) a file from a source path to a destination path. The move operation, like the copy operation, requires no actual transfer of data, and is performed as follows:

(A) Copy the file from the source path to the destination path.

(B) If the source path is different from the destination path, delete the source path.

8. Get File Status

This mechanism takes a file pathname and provides information about the pathname. First the local directory extensions table entry record **138** corresponding to the pathname given is found. If no such entry exists, then this mechanism fails, otherwise, gather information about the file and its corresponding True File from the local directory extensions table **124**. The information can include any information shown in the data structures, including the size, type, owner, True Name, sources, time of last access, time of last modification, state (local or not, assimilated or not, compressed or not), use count, expiration date, and reservations.

9. Get Files in Directory

This mechanism enumerates the files in a directory. It is used (implicitly) whenever it is necessary to determine whether a file exists (is present) in a directory. For instance, it is implicitly used in the Open File, Delete File, Copy File or Directory, and Move File operating system mechanisms, because the files operated on are referred to by pathnames containing directory names. The mechanism works as follows:

The local directory extensions table **124** is searched for an entry **138** with the given directory pathname. If no such entry is found, or if the entry found is not a directory, then this mechanism fails.

If there is a corresponding True File field in the local directory extensions table record, then it is assumed that the True File represents a frozen directory. The Expand Frozen Directory primitive mechanism is used to expand the existing True File into directory entries in the local directory extensions table.

Finally, the local directory extensions table **124** is again searched, this time to find each directory subordinate to the given directory. The names found are provided as the result.

Remote Mechanisms

The remote mechanisms provided by the present invention are now described. Recall that remote mechanisms are used by the operating system in responding to requests from other processors. These mechanisms enable the capabilities of the present invention in a peer-to-peer network mode of operation.

In a presently preferred embodiment, processors communicate with each other using a remote procedure call (RPC) style interface, running over one of any number of communication protocols such as IPX/SPX or TCP/IP. Each peer processor which provides access to its True File registry **126** or file regions, or which depends on another peer processor, provides a number of mechanisms which can be used by its peers.

The following remote mechanisms are described:

1. Locate True File;
2. Reserve True File;
3. Request True File;
4. Retire True File;
5. Cancel Reservation;
6. Acquire True File;
7. Lock Cache;
8. Update Cache; and
9. Check Expiration Date.

1. Locate True File

This mechanism allows a remote processor to determine whether the local processor contains a copy of a specific True File. The mechanism begins with a True Name and a flag indicating whether to forward requests for this file to other servers. This mechanism is now described with reference to FIG. **28**.

First determine if the True File is available locally or if there is some indication of where the True File is located (for example, in the Source IDs field). Look up the requested True Name in the True File registry **126** (Step **S432**).

If a True File registry entry record **140** is not found for this True Name (Step **S434**), and the flag indicates that the request is not to be forwarded (Step **S436**), respond negatively (Step **S438**). That is, respond to the effect that the True File is not available.

One the other hand, if a True File registry entry record **140** is not found (Step **S434**), and the flag indicates that the request for this True File is to be forwarded (Step **S436**), then forward a request for this True File to some other processors in the system (Step **S442**). If the source table for the current processor identifies one or more publishing servers which should have a copy of this True File, then forward the request to each of those publishing servers (Step **S436**).

If a True File registry entry record **140** is found for the required True File (Step **S434**), and if the entry includes a True File ID or Compressed File ID (Step **S440**), respond positively (Step **S444**). If the entry includes a True File ID then this provides the identity or disk location of the actual physical representation of the file or file segment required. If the entry include a Compressed File ID, then a compressed version of the True File may be stored instead of, or in addition to, an uncompressed version. This field provides the identity of the actual representation of the compressed version of the file.

If the True File registry entry record **140** is found (Step **S434**) but does not include a True File ID (the File ID is absent if the actual file is not currently present at the current location) (Step **S440**), and if the True File registry entry record **140** includes one or more source processors, and if the request can be forwarded, then forward the request for this True File to one or more of the source processors (Step **S444**).

2. Reserve True File

This mechanism allows a remote processor to indicate that it depends on the local processor for access to a specific True File. It takes a True Name as input. This mechanism is described here.

(A) Find the True File registry entry record **140** associated with the given True File. If no entry exists, reply negatively.

(B) If the True File registry entry record **140** does not include a True File ID or compressed File ID, and if the True File registry entry record **140** includes no source IDs for removable storage volumes, then this processor does not have access to a copy of the given file. Reply negatively.

(C) Add the ID of the sending processor to the list of dependent processors for the True File registry entry record **140**. Reply positively, with an indication of whether the reserved True File is on line or off line.

3. Request True File

This mechanism allows a remote processor to request a copy of a True File from the local processor. It requires a True Name and responds positively by sending a True File back to the requesting processor. The mechanism operates as follows:

(A) Find the True File registry entry record **140** associated with the given True Name. If there is no such True File registry entry record 140, reply negatively.

(B) Make the True File local using the Make True File Local primitive mechanism. If this mechanism fails, the Request True File mechanism also fails.

(C) Send the local True File in either it is uncompressed or compressed form to the requesting remote processor. Note that if the True File is a compound file, the components are not sent.

(D) If the remote file is listed in the dependent process list of the True File registry entry record **140**, remove it.

4. Retire True File

This mechanism allows a remote processor to indicate that it no longer plans to maintain a copy of a given True File. An alternate source of the True File can be specified, if, for instance, the True File is being moved from one server to another. It begins with a True Name, a requesting processor ID, and an optional alternate source. This mechanism operates as follows:

(A) Find a True Name entry in the True File registry **126**. If there is no entry for this True Name, this mechanism's task is complete.

(B) Find the requesting processor on the source list and, if it is there, remove it.

(C) If an alternate source is provided, add it to the source list for the True File registry entry record **140**.

(D) If the source list of the True File registry entry record **140** has no items in it, use the Locate Remote File primitive mechanism to search for another copy of the file. If it fails, raise a serious error.

5. Cancel Reservation

This mechanism allows a remote processor to indicate that it no longer requires access to a True File stored on the local processor. It begins with a True Name and a requesting processor ID and proceeds as follows:

(A) Find the True Name entry in the True File registry **126**. If there is no entry for this True Name, this mechanism's task is complete.

(B) Remove the identity of the requesting processor from the list of dependent processors, if it appears.

(C) If the list of dependent processors becomes zero and the use count is also zero, delete the True File.

6. Acquire True File

This mechanism allows a remote processor to insist that a local processor make a copy of a specified True File. It is used, for example, when a cache client wants to write through a new version of a file. The Acquire True File mechanism begins with a data item and an optional True Name for the data item and proceeds as follows:

(A) Confirm that the requesting processor has the right to require the local processor to acquire data items. If not, send a negative reply.

(B) Make a local copy of the data item transmitted by the remote processor.

(C) Assimilate the data item into the True File registry of the local processor.

(D) If a True Name was provided with the file, the True Name calculation can be avoided, or the mechanism can verify that the file received matches the True Name sent.

(E) Add an entry in the dependent processor list of the true file registry record indicating that the requesting processor depends on this copy of the given True File.

(F) Send a positive reply.

7. Lock Cache

This mechanism allows a remote cache client to lock a local file so that local users or other cache clients cannot change it while the remote processor is using it. The mechanism begins with a pathname and proceeds as follows:

(A) Find the local directory extensions table entry record **138** of the specified pathname. If no such entry exists, reply negatively.

(B) If an local directory extensions table entry record **138** exists and is already locked, reply negatively that the file is already locked.

(C) If an local directory extensions table entry record **138** exists and is not locked, lock the entry. Reply positively.

8. Update Cache

This mechanism allows a remote cache client to unlock a local file and update it with new contents. It begins with a pathname and a True Name. The file corresponding to the True Name must be accessible from the remote processor. This mechanism operates as follows:

Find the local directory extensions table entry record **138** corresponding to the given pathname. Reply negatively if no such entry exists or if the entry is not locked.

Link the given pathname to the given True Name using the Link Path to True Name primitive mechanism.

Unlock the local directory extensions table entry record **138** and return positively.

9. Check Expiration Date

Return current or new expiration date and possible alternative source to caller.

Background Processes and Mechanisms

The background processes and mechanisms provided by the present invention are now described. Recall that background mechanisms are intended to run occasionally and at a low priority to provide automated management capabilities with respect to the present invention.

The following background mechanisms are described:

1. Mirror True File;

2. Groom Region;

3. Check for Expired Links;

4. Verify Region; and

5. Groom Source List.

1. Mirror True File

This mechanism is used to ensure that files are available in alternate locations in mirror groups or archived on archival servers. The mechanism depends on application-specific migration/archival criteria (size, time since last access, number of copies required, number of existing alternative sources) which determine under what conditions a file should be moved. The Mirror True File mechanism operates as follows, using the True File specified, perform the following steps:

(A) Count the number of available locations of the True File by inspecting the source list of the True File registry entry record **140** for the True File. This step determines how many copies of the True File are available in the system.

(B) If the True File meets the specified migration criteria, select a mirror group server to which a copy of the file should be sent. Use the Acquire True File remote mechanism to copy the True File to the selected mirror group server. Add the identity of the selected system to the source list for the True File.

2. Groom Region

This mechanism is used to automatically free up space in a processor by deleting data items that may be available elsewhere. The mechanism depends on application-specific grooming criteria (for instance, a file may be removed if there is an alternate online source for it, it has not been accessed in a given number of days, and it is larger than a given size). This mechanism operates as follows:

Repeat the following steps (i) to (iii) with more aggressive grooming criteria until sufficient space is freed or until all grooming criteria have been exercised. Use grooming information to determine how much space has been freed. Recall that, while grooming is in effect, grooming information includes a table of pathnames selected for deletion, and keeps track of the amount of space that would be freed if all of the files were deleted.

(i) Begin Grooming (using the primitive mechanism).

(ii) For each pathname in the specified region, for the True File corresponding to the pathname, if the True File is present, has at least one alternative source, and meets application specific grooming criteria for the region, select the file for removal (using the primitive mechanism).

(iii) End Grooming (using the primitive mechanism).

If the region is used as a cache, no other processors are dependent on True Files to which it refers, and all such True Files are mirrored elsewhere. In this case, True Files can be removed with impunity. For a cache region, the grooming criteria would ordinarily eliminate the least recently accessed True Files first. This is best done by sorting the True Files in the region by the most recent access time before performing step (ii) above. The application specific criteria would thus be to select for removal every True File encountered (beginning with the least recently used) until the required amount of free space is reached.

3. Check for Expired Links

This mechanism is used to determine whether dependencies on published files should be refreshed. The following steps describe the operation of this mechanism:

For each pathname in the specified region, for each True File corresponding to the pathname, perform the following step:

If the True File registry entry record **140** corresponding to the True File contains at least one source which is a publishing server, and if the expiration date on the dependency is past or close, then perform the following steps:

(A) Determine whether the True File registry entry record contains other sources which have not expired.

(B) Check the True Name expiration of the server. If the expiration date has been extended, or an alternate source is suggested, add the source to the True File registry entry record **140**.

(C) If no acceptable alternate source was found in steps (A) or (B) above, make a local copy of the True File.

(D) Remove the expired source.

4. Verify Region

This mechanism can be used to ensure that the data items in the True File registry **126** have not been damaged accidentally or maliciously. The operation of this mechanism is described by the following steps:

(A) Search the local directory extensions table **124** for each pathname in the specified region and then perform the following steps:

(i) Get the True File name corresponding to the pathname;

(ii) If the True File registry entry **140** for the True File does not have a True File ID or compressed file ID, ignore it.

(iii) Use the Verify True File mechanism (see extended mechanisms below) to confirm that the True File specified is correct.

5. Groom Source List

The source list in a True File entry should be groomed sometimes to ensure there are not too many mirror or archive copies. When a file is deleted or when a region definition or its mirror criteria are changed, it may be necessary to inspect the affected True Files to determine whether there are too many mirror copies. This can be done with the following steps:

For each affected True File,

(A) Search the local directory extensions table to find each region that refers to the True File.

(B) Create a set of "required sources", initially empty.

(C) For each region found,

(a) determine the mirroring criteria for that region,

(b) determine which sources for the True File satisfy the mirroring criteria, and

(c) add these sources to the set of required sources.

(D) For each source in the True File registry entry, if the source identifies a remote processor (as opposed to removable media), and if the source is not a publisher, and if the source is not in the set of required sources, then eliminate the source, and use the Cancel Reservation remote mechanism to eliminate the given processor from the list of dependent processors recorded at the remote processor identified by the source.

Extended Mechanisms

The extended mechanisms provided by the present invention are now described. Recall that extended mechanisms run within application programs over the operating system to provide solutions to specific problems and applications.

The following extended mechanisms are described:

1. Inventory Existing Directory;
2. Inventory Removable, Read-only Files;
3. Synchronize Directories;
4. Publish Region;
5. Retire Directory;
6. Realize Directory at Location;
7. Verify True File;
8. Track for Accounting Purposes; and
9. Track for Licensing Purposes.

1. Inventory Existing Directory

This mechanism determines the True Names of files in an existing on-line directory in the underlying operating system. One purpose of this mechanism is to install True Name mechanisms in an existing file system.

An effect of such an installation is to eliminate immediately all duplicate files from the file system being traversed. If several file systems are inventoried in a single True File registry, duplicates across the volumes are also eliminated.

(A) Traverse the underlying file system in the operating system. For each file encountered, excluding directories, perform the following:

(i) Assimilate the file encountered (using the Assimilate File primitive mechanism). This process computes its True Name and moves its data into the True File registry **126**.

(ii) Create a pathname consisting of the path to the volume directory and the relative path of the file on the media. Link this path to the computed True Name using the Link Path to True Name primitive mechanism.

2. Inventory Removable, Read-only Files

A system with access to removable, read-only media volumes (such as WORM disks and CD-ROMs) can create a usable inventory of the files on these disks without having to make online copies. These objects can then be used for archival purposes, directory overlays, or other needs. An operator must request that an inventory be created for such a volume.

This mechanism allows for maintaining inventories of the contents of files and data items on removable media, such as diskettes and CD-ROMS, independent of other properties of the files such as name, location, and date of creation.

The mechanism creates an online inventory of the files on one or more removable volumes, such as a floppy disk or CD-ROM, when the data on the volume is represented as a directory. The inventory service uses a True Name to identify each file, providing a way to locate the data independent of its name, date of creation, or location.

The inventory can be used for archival of data (making it possible to avoid archiving data when that data is already on a separate volume), for grooming (making it possible to delete infrequently accessed files if they can be retrieved from removable volumes), for version control (making it possible to generate a new version of a CD-ROM without having to copy the old version), and for other purposes.

The inventory is made by creating a volume directory in the media inventory in which each file named identifies the data item on the volume being inventoried. Data items are not copied from the removable volume during the inventory process.

An operator must request that an inventory be created for a specific volume. Once created, the volume directory can be frozen or copied like any other directory. Data items from either the physical volume or the volume directory can be accessed using the Open File operating system mechanism which will cause them to be read from the physical volume using the Realize True File from Location primitive mechanism.

To create an inventory the following steps are taken:

(A) A volume directory in the media inventory is created to correspond to the volume being inventoried. Its contextual name identifies the specific volume.

(B) A source table entry **144** for the volume is created in the source table **130**. This entry **144** identifies the physical source volume and the volume directory created in step (A).

(C) The filesystem on the volume is traversed. For each file encountered, excluding directories, the following steps are taken:

(i) The True Name of the file is computed. An entry is created in the True Name registry **124**, including the True Name of the file using the primitive mechanism. The source field of the True Name registry entry **140** identifies the source table entry **144**.

(ii) A pathname is created consisting of the path to the volume directory and the relative path of the file on the media. This path is linked to the computed True Name using Link Path to True Name primitive mechanism.

(D) After all files have been inventoried, the volume directory is frozen. The volume directory serves as a table of contents for the volume. It can be copied using the Copy File or Directory primitive mechanism to create an "overlay" directory which can then be modified, making it possible to edit a virtual copy of a read-only medium.

3. Synchronize Directories

Given two versions of a directory derived from the same starting point, this mechanism creates a new, synchronized version which includes the changes from each. Where a file is changed in both versions, this mechanism provides a user exit for handling the discrepancy. By using True Names, comparisons are instantaneous, and no copies of files are necessary.

This mechanism lets a local processor synchronize a directory to account for changes made at a remote processor. Its purpose is to bring a local copy of a directory up to date after a period of no communication between the local and remote processor. Such a period might occur if the local processor were a mobile processor detached from its server, or if two distant processors were run independently and updated nightly.

An advantage of the described synchronization process is that it does not depend on synchronizing the clocks of the local and remote processors. However, it does require that the local processor track its position in the remote processor's audit file.

This mechanism does not resolve changes made simultaneously to the same file at several sites. If that occurs, an external resolution mechanism such as, for example, operator intervention, is required.

The mechanism takes as input a start time, a local directory pathname, a remote processor name, and a remote directory pathname name, and it operates by the following steps:

(A) Request a copy of the audit file **132** from the remote processor using the Request True File remote mechanism.

(B) For each entry **146** in the audit file **132** after the start time, if the entry indicates a change to a file in the remote directory, perform the following steps:

(i) Compute the pathname of the corresponding file in the local directory. Determine the True Name of the corresponding file.

(ii) If the True Name of the local file is the same as the old True Name in the audit file, or if there is no local file and the audit entry indicates a new file is being created, link the new True Name in the audit file to the local pathname using the Link Path to True Name primitive mechanism.

(iii) Otherwise, note that there is a problem with the synchronization by sending a message to the operator or to a problem resolution program, indicating the local pathname, remote pathname, remote processor, and time of change.

(C) After synchronization is complete, record the time of the final change. This time is to be used as the new start time the next time this directory is synchronized with the same remote processor.

4. Publish Region

The publish region mechanism allows a processor to offer the files in a region to any client processors for a limited period of time.

The purpose of the service is to eliminate any need for client processors to make reservations with the publishing processor. This in turn makes it possible for the publishing processor to service a much larger number of clients.

When a region is published, an expiration date is defined for all files in the region, and is propagated into the publishing system's True File registry entry record **140** for each file.

When a remote file is copied, for instance using the Copy File operating system mechanism, the expiration date is copied into the source field of the client's True File registry entry record **140**. When the source is a publishing system, no dependency need be created.

The client processor must occasionally and in background, check for expired links, to make sure it still has access to these files. This is described in the background mechanism Check for Expired Links.

5. Retire Directory

This mechanism makes it possible to eliminate safely the True Files in a directory, or at least dependencies on them, after ensuring that any client processors depending on those files remove their dependencies. The files in the directory are not actually deleted by this process. The directory can be deleted with the Delete File operating system mechanism.

The mechanism takes the pathname of a given directory, and optionally, the identification of a preferred alternate source processor for clients to use. The mechanism performs the following steps:

(A) Traverse the directory. For each file in the directory, perform the following steps:

(i) Get the True Name of the file from its path and find the True File registry entry **140** associated with the True Name.

(ii) Determine an alternate source for the True File. If the source IDs field of the TFR entry includes the preferred alternate source, that is the alternate source. If it does not, but includes some other source, that is the alternate source. If it contains no alternate sources, there is no alternate source.

(iii) For each dependent processor in the True File registry entry **140**, ask that processor to retire the True File, specifying an alternate source if one was determined, using the remote mechanism.

6. Realize Directory at Location

This mechanism allows the user or operating system to force copies of files from some source location to the True File registry **126** at a given location. The purpose of the mechanism is to ensure that files are accessible in the event the source location becomes inaccessible. This can happen for instance if the source or given location are on mobile computers, or are on removable media, or if the network connection to the source is expected to become unavailable, or if the source is being retired.

This mechanism is provided in the following steps for each file in the given directory, with the exception of subdirectories:

(A) Get the local directory extensions table entry record **138** given the pathname of the file. Get the True Name of the local directory extensions table entry record **138**. This service assimilates the file if it has not already been assimilated.

(B) Realize the corresponding True File at the given location. This service causes it to be copied to the given location from a remote system or removable media.

7. Verify True File

This mechanism is used to verify that the data item in a True File registry **126** is indeed the correct data item given its True Name. Its purpose is to guard against device errors, malicious changes, or other problems.

If an error is found, the system has the ability to "heal" itself by finding another source for the True File with the given name. It may also be desirable to verify that the error has not propagated to other systems, and to log the problem or indicate it to the computer operator. These details are not described here.

To verify a data item that is not in a True File registry **126**, use the Calculate True Name primitive mechanism described above.

The basic mechanism begins with a True Name, and operates in the following steps:

(A) Find the True File registry entry record **140** corresponding to the given True Name.

(B) If there is a True File ID for the True File registry entry record **140** then use it. Otherwise, indicate that no file exists to verify.

(C) Calculate the True Name of the data item given the file ID of the data item.

(D) Confirm that the calculated True Name is equal to the given True Name.

(E) If the True Names are not equal, there is an error in the True File registry **126**. Remove the True File ID from the True File registry entry record **140** and place it somewhere else. Indicate that the True File registry entry record **140** contained an error.

8. Track for Accounting Purposes

This mechanism provides a way to know reliably which files have been stored on a system or transmitted from one system to another. The mechanism can be used as a basis for a value-based accounting system in which charges are based on the identity of the data stored or transmitted, rather than simply on the number of bits.

This mechanism allows the system to track possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes. True names make it possible to identify each file briefly yet uniquely for this purpose.

Tracking the identities of files requires maintaining an accounting log **134** and processing it for accounting or billing purposes. The mechanism operates in the following steps:

(A) Note every time a file is created or deleted, for instance by monitoring audit entries in the Process Audit File Entry primitive mechanism. When such an event is encountered, create an entry **148** in the accounting log **134** that shows the responsible party and the identity of the file created or deleted.

(B) Every time a file is transmitted, for instance when a file is copied with a Request True File remote mechanism or an Acquire True File remote mechanism, create an entry in the accounting log **134** that shows the responsible party, the identity of the file, and the source and destination processors.

(C) Occasionally run an accounting program to process the accounting log **134**, distributing the events to the account records of each responsible party. The account records can eventually be summarized for billing purposes.

9. Track for Licensing Purposes

This mechanism ensures that licensed files are not used by unauthorized parties. The True Name provides a safe way to identify licensed material. This service allows proof of possession of specific files according to their contents without disclosing their contents.

Enforcing use of valid licenses can be active (for example, by refusing to provide access to a file without authorization) or passive (for example, by creating a report of users who do not have proper authorization).

One possible way to perform license validation is to perform occasional audits of employee systems. The service described herein relies on True Names to support such an audit, as in the following steps:

(A) For each licensed product, record in the license table **136** the True Name of key files in the product (that is, files which are required in order to use the product, and which do not occur in other products) Typically, for a software product, this would include the main executable image and perhaps other major files such as clip-art, scripts, or online help. Also record the identity of each system which is authorized to have a copy of the file.

(B) Occasionally, compare the contents of each user processor against the license table **136**. For each True Name in the license table do the following:

(i) Unless the user processor is authorized to have a copy of the file, confirm that the user processor does not have a copy of the file using the Locate True File mechanism.

(ii) If the user processor is found to have a file that it is not authorized to have, record the user processor and True Name in a license violation table.

The System in Operation

Given the mechanisms described above, the operation of a typical DP system employing these mechanisms is now described in order to demonstrate how the present invention meets its requirements and capabilities.

In operation, data items (for example, files, database records, messages, data segments, data blocks, directories, instances of object classes, and the like) in a DP system employing the present invention are identified by substantially unique identifiers (True Names), the identifiers depending on all of the data in the data items and only on the data in the data items. The primitive mechanisms Calculate True Name and Assimilate Data Item support this property. For any given data item, using the Calculate True Name primitive mechanism, a substantially unique identifier or True Name for that data item can be determined.

Further, in operation of a DP system incorporating the present invention, multiple copies of data items are avoided (unless they are required for some reason such as backups or mirror copies in a fault-tolerant system). Multiple copies of data items are avoided even when multiple names refer to the same data item. The primitive mechanisms Assimilate Data Items and New True File support this property. Using the Assimilate Data Item primitive mechanism, if a data item already exists in the system, as indicated by an entry in the True File registry **126**, this existence will be discovered by this mechanism, and the duplicate data item (the new data item) will be eliminated (or not added). Thus, for example, if a data file is being copied onto a system from a floppy disk, if, based on the True Name of the data file, it is determined that the data file already exists in the system (by the same or some other name), then the duplicate copy will not be installed. If the data item was being installed on the system by some name other than its current name, then, using the Link Path to True Name primitive mechanism, the other (or new) name can be linked to the already existing data item.

In general, the mechanisms of the present invention operate in such a way as to avoid recreating an actual data item at a location when a copy of that data item is already present at that location. In the case of a copy from a floppy disk, the data item (file) may have to be copied (into a scratch file) before it can be determined that it is a duplicate. This is because only one processor is involved. On the other hand, in a multiprocessor environment or DP system, each processor has a record of the True Names of the data items on that processor. When a data item is to be copied to another location (another processor) in the DP system, all that is necessary is to examine the True Name of the data item prior to the copying. If a data item with the same True Name already exists at the destination location (processor), then there is no need to copy the data item. Note that if a data item which already exists locally at a destination location is still copied to the destination location (for example, because the remote system did not have a True Name for the data item or because it arrives as a stream of un-named data), the Assimilate Data Item primitive mechanism will prevent multiple copies of the data item from being created.

Since the True Name of a large data item (a compound data item) is derived from and based on the True Names of components of the data item, copying of an entire data item can be avoided. Since some (or all) of the components of a large data item may already be present at a destination location, only those components which are not present there need be copied. This property derives from the manner in which True Names are determined.

When a file is copied by the Copy File or Directory operating system mechanism, only the True Name of the file is actually replicated.

When a file is opened (using the Open File operating system mechanism), it uses the Make True File Local primitive mechanism (either directly or indirectly through the Create Scratch File primitive mechanism) to create a local copy of the file. The Open File operating system mechanism uses the Make True File Local primitive mechanism, which uses the Realize True File from Location primitive mechanism, which, in turn uses the Request True File remote mechanism.

The Request True File remote mechanism copies only a single data item from one processor to another. If the data item is a compound file, its component segments are not copied, only the indirect block is copied. The segments are copied only when they are read (or otherwise needed).

The Read File operating system mechanism actually reads data. The Read File mechanism is aware of compound files and indirect blocks, and it uses the Realize True File from Location primitive mechanism to make sure that component segments are locally available, and then uses the operating system file mechanisms to read data from the local file.

Thus, when a compound file is copied from a remote system, only its True Name is copied. When it is opened, only its indirect block is copied. When the corresponding file is read, the required component segments are realized and therefore copied.

In operation data items can be accessed by reference to their identities (True Names) independent of their present location. The actual data item or True File corresponding to a given data identifier or True Name may reside anywhere in the system (that is, locally, remotely, offline, etc). If a required True File is present locally, then the data in the file can be accessed. If the data item is not present locally, there are a number of ways in which it can be obtained from wherever it is present. Using the source IDs field of the True File registry table, the location(s) of copies of the True File corresponding to a given True Name can be determined. The Realize True File from Location primitive mechanism tries to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File. If, on the other hand, for some reason it is not known where there is a copy of the True File, or if the processors identified in the source IDs field do not respond with the required True File, the processor requiring the data item can make a general request for the data item using the Request True File remote mechanism from all processors in the system that it can contact.

As a result, the system provides transparent access to any data item by reference to its data identity, and independent of its present location.

In operation, data items in the system can be verified and have their integrity checked. This is from the manner in which True Names are determined. This can be used for security purposes, for instance, to check for viruses and to verify that data retrieved from another location is the desired and requested data. For example, the system might store the True Names of all executable applications on the system and then periodically redetermine the True Names of each of these applications to ensure that they match the stored True Names. Any change in a True Name potentially signals corruption in the system and can be further investigated. The Verify Region background mechanism and the Verify True File extended mechanisms provide direct support for this mode of operation. The Verify Region mechanism is used to ensure that the data items in the True File registry have not been damaged accidentally or maliciously. The Verify True File mechanism verifies that a data item in a True File registry is indeed the correct data item given its True Name.

Once a processor has determined where (that is, at which other processor or location) a copy of a data item is in the DP system, that processor might need that other processor or location to keep a copy of that data item. For example, a processor might want to delete local copies of data items to make space available locally while knowing that it can rely on retrieving the data from somewhere else when needed. To this end the system allows a processor to Reserve (and cancel the reservation of) True Files at remote locations (using the remote mechanism). In this way the remote locations are put on notice that another location is relying on the presence of the True File at their location.

A DP system employing the present invention can be made into a fault-tolerant system by providing a certain amount of redundancy of data items at multiple locations in the system. Using the Acquire True File and Reserve True File remote mechanisms, a particular processor can implement its own form of fault-tolerance by copying data items to other processors and then reserving them there. However, the system also provides the Mirror True File background mechanism to mirror (make copies) of the True File available elsewhere in the system. Any degree of redundancy (limited by the number of processors or locations in the system) can be implemented. As a result, this invention maintains a desired degree or level of redundancy in a network of processors, to protect against failure of any particular processor by ensuring that multiple copies of data items exist at different locations.

The data structures used to implement various features and mechanisms of this invention store a variety of useful information which can be used, in conjunction with the various mechanisms, to implement storage schemes and policies in a DP system employing the invention. For example, the size, age and location of a data item (or of groups of data items) is provided. This information can be used to decide how the data items should be treated. For example, a processor may implement a policy of deleting local copies of all data items over a certain age if other copies of those data items are present elsewhere in the system. The age (or variations on the age) can be determined using the time of last access or modification in the local directory extensions table, and the presence of other copies of the data item can be determined either from the Safe Flag or the source IDs, or by checking which other processors in the system have copies of the data item and then reserving at least one of those copies.

In operation, the system can keep track of data items regardless of how those items are named by users (or regardless of whether the data items even have names). The system can also track data items that have different names (in different or the same location) as well as different data items that have the same name. Since a data item is identified by the data in the item, without regard for the context of the data, the problems of inconsistent naming in a DP system are overcome.

In operation, the system can publish data items, allowing other, possibly anonymous, systems in a network to gain access to the data items and to rely on the availability of these data items. True Names are globally unique identifiers which can be published simply by copying them. For example, a user might create a textual representation of a file on system A with True Name N (for instance as a hexadecimal string), and post it on a computer bulletin board. Another user on system B could create a directory entry F for this True Name N by using the Link Path to True Name primitive mechanism. (Alternatively, an application could be developed which hides the True Name from the users, but provides the same public transfer service.)

When a program on system B attempts to open pathname F linked to True Name N, the Locate Remote File primitive mechanism would be used, and would use the Locate True File remote mechanism to search for True Name N on one or more remote processors, such as system A. If system B has access to system A, it would be able to realize the True File (using the Realize True File from Location primitive mechanism) and use it locally. Alternatively, system B could find True Name N by accessing any publicly available True Name server, if the server could eventually forward the request to system A.

Clients of a local server can indicate that they depend on a given True File (using the Reserve True File remote mechanism) so that the True File is not deleted from the server registry as long as some client requires access to it. (The Retire True File remote mechanism is used to indicate that a client no longer needs a given True File.)

A publishing server, on the other hand, may want to provide access to many clients, and possibly anonymous ones, without incurring the overhead of tracking dependencies for each client. Therefore, a public server can provide expiration dates for True Files in its registry. This allows client systems to safely maintain references to a True File on the public server. The Check For Expired Links background mechanism allows the client of a publishing server to occasionally confirm that its dependencies on the publishing server are safe.

In a variation of this aspect of the invention, a processor that is newly connected (or reconnected after some absence) to the system can obtain a current version of all (or of needed) data in the system by requesting it from a server processor. Any such processor can send a request to update or resynchronize all of its directories (starting at a root directory), simply by using the Synchronize Directories extended mechanism on the needed directories.

Using the accounting log or some other user provided mechanism, a user can prove the existence of certain data items at certain times. By publishing (in a public place) a list of all True Names in the system on a given day (or at some given time), a user can later refer back to that list to show that a particular data item was present in the system at the time that list was published. Such a mechanism is useful in tracking, for example, laboratory notebooks or the like to prove dates of conception of inventions. Such a mechanism also permits proof of possession of a data item at a particular date and time.

The accounting log file can also track the use of specific data items and files by content for accounting purposes. For instance, an information utility company can determine the data identities of data items that are stored and transmitted through its computer systems, and use these identities to provide bills to its customers based on the identities of the data items being transmitted (as defined by the substantially unique identifier). The assignment of prices for storing and transmitting specific True Files would be made by the information utility and/or its data suppliers; this information would be joined periodically with the information in the accounting log file to produce customer statements.

Backing up data items in a DP system employing the present invention can be done based on the True Names of the data items. By tracking backups using True Names, duplication in the backups is prevented. In operation, the system maintains a backup record of data identifiers of data items already backed up, and invokes the Copy File or Directory operating system mechanism to copy only those data items whose data identifiers are not recorded in the backup record. Once a data item has been backed up, it can be restored by retrieving it from its backup location, based on the identifier of the data item. Using the backup record produced by the backup to identify the data item, the data item can be obtained using, for example, the Make True File Local primitive mechanism.

In operation, the system can be used to cache data items from a server, so that only the most recently accessed data items need be retained. To operate in this way, a cache client is configured to have a local registry (its cache) with a remote Local Directory Extensions table (from the cache server). Whenever a file is opened (or read), the Local Directory Extensions table is used to identify the True Name, and the Make True File Local primitive mechanism inspects the local registry. When the local registry already has a copy, the file is already cached. Otherwise, the Locate True File remote mechanism is used to get a copy of the file. This mechanism consults the cache server and uses the

Request True File remote mechanism to make a local copy, effectively loading the cache.

The Groom Cache background mechanism flushes the cache, removing the least-recently-used files from the cache client's True File registry. While a file is being modified on a cache client, the Lock Cache and Update Cache remote mechanisms prevent other clients from trying to modify the same file.

In operation, when the system is being used to cache data items, the problems of maintaining cache consistency are avoided.

To access a cache and to fill it from its server, a key is required to identify the data item desired. Ordinarily, the key is a name or address (in this case, it would be the pathname of a file). If the data associated with such a key is changed, the client's cache becomes inconsistent; when the cache client refers to that name, it will retrieve the wrong data. In order to maintain cache consistency it is necessary to notify every client immediately whenever a change occurs on the server.

By using an embodiment of the present invention, the cache key uniquely identifies the data it represents. When the data associated with a name changes, the key itself changes. Thus, when a cache client wishes to access the modified data associated with a given file name, it will use a new key (the True Name of the new file) rather than the key to the old file contents in its cache. The client will always request the correct data, and the old data in its cache will be eventually aged and flushed by the Groom Cache background mechanism.

Because it is not necessary to immediately notify clients when changes on the cache server occur, the present invention makes it possible for a single server to support a much larger number of clients than is otherwise possible.

In operation, the system automatically archives data items as they are created or modified. After a file is created or modified, the Close File operating system mechanism creates an audit file record, which is eventually processed by the Process Audit File Entry primitive mechanism. This mechanism uses the New True File primitive mechanism for any file which is newly created, which in turn uses the Mirror True File background mechanism if the True File is in a mirrored or archived region. This mechanism causes one or more copies of the new file to be made on remote processors.

In operation, the system can efficiently record and preserve any collection of data items. The Freeze Directory primitive mechanism creates a True File which identifies all of the files in the directory and its subordinates. Because this True File includes the True Names of its constituents, it represents the exact contents of the directory tree at the time it was frozen. The frozen directory can be copied with its components preserved.

The Acquire True File remote mechanism (used in mirroring and archiving) preserves the directory tree structure by ensuring that all of the component segments and True Files in a compound data item are actually copied to a remote system. Of course, no transfer is necessary for data items already in the registry of the remote system.

In operation, the system can efficiently make a copy of any collection of data items, to support a version control mechanism for groups of the data items.

The Freeze Directory primitive mechanism is used to create a collection of data items. The constituent files and segments referred to by the frozen directory are maintained in the registry, without any need to make copies of the constituents each time the directory is frozen.

Whenever a pathname is traversed, the Get Files in Directory operating system mechanism is used, and when it encounters a frozen directory, it uses the Expand Frozen Directory primitive mechanism.

A frozen directory can be copied from one pathname to another efficiently, merely by copying its True Name. The Copy File operating system mechanism is used to copy a frozen directory.

Thus it is possible to efficiently create copies of different versions of a directory, thereby creating a record of its history (hence a version control system).

In operation, the system can maintain a local inventory of all the data items located on a given removable medium, such as a diskette or CD-ROM. The inventory is independent of other properties of the data items such as their name, location, and date of creation.

The Inventory Existing Directory extended mechanism provides a way to create True File Registry entries for all of the files in a directory. One use of this inventory is as a way to pre-load a True File registry with backup record information. Those files in the registry (such as previously installed software) which are on the volumes inventoried need not be backed up onto other volumes.

The Inventory Removable, Read-only Files extended mechanism not only determines the True Names for the files on the medium, but also records directory entries for each file in a frozen directory structure. By copying and modifying this directory, it is possible to create an on line patch, or small modification of an existing read-only file. For example, it is possible to create an online representation of a modified CD-ROM, such that the unmodified files are actually on the CD-ROM, and only the modified files are online.

In operation, the system tracks possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes. Using the Track for Accounting Purposes extended mechanism provides a way to know reliably which files have been stored on a system or transmitted from one system to another.

True Names in Relational and Object-Oriented Databases

Although the preferred embodiment of this invention has been presented in the context of a file system, the invention of True Names would be equally valuable in a relational or object-oriented database. A relational or object-oriented database system using True Names would have similar benefits to those of the file system employing the invention. For instance, such a database would permit efficient elimination of duplicate records, support a cache for records, simplify the process of maintaining cache consistency, provide location-independent access to records, maintain archives and histories of records, and synchronize with distant or disconnected systems or databases.

The mechanisms described above can be easily modified to serve in such a database environment. The True Name registry would be used as a repository of database records. All references to records would be via the True Name of the record. (The Local Directory Extensions table is an example of a primary index that uses the True Name as the unique identifier of the desired records.)

In such a database, the operations of inserting, updating, and deleting records would be implemented by first assimilating records into the registry, and then updating a primary key index to map the key of the record to its contents by using the True Name as a pointer to the contents.

The mechanisms described in the preferred embodiment, or similar mechanisms, would be employed in such a system. These mechanisms could include, for example, the mechanisms for calculating true names, assimilating, locating, realizing, deleting, copying, and moving True Files, for mirroring True Files, for maintaining a cache of True Files, for grooming True Files, and other mechanisms based on the use of substantially unique identifiers.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiments, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

What is claimed is:

1. In a data processing system, an apparatus comprising:

identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

2. An apparatus as in claim 1, further comprising:

local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item.

3. An apparatus as in claim 2, wherein each location contains a distinct plurality of data items, and wherein said local existence means determines whether a particular data item is present at a particular location in the system by examining the identifiers of the plurality of data items at said particular location in the system.

4. An apparatus as in claim 2, further comprising:

data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item; and

access means for accessing a particular data item using the identifier of the data item.

5. An apparatus as in claim 2, further comprising:

duplication means for copying a data item from a source to a destination in the data processing system, by providing said destination with the data item only if it is determined using the data identifier that the data item is not present at the destination.

6. An apparatus as in claim 4, further comprising:

assimilation means for assimilating a new data item into the system, said assimilation means invoking said identity means to determine the identifier of the new data item and invoking said data associating means to associate the new data item with its identifier.

7. An apparatus as in claim 4, further comprising:

duplication means for duplicating a data item from a source location to a destination location in the data processing system, based on the identifier of the data item, said duplication means invoking said local existence means to determine whether an instance of the data item is present at the destination location, and invoking said access means to provide said destination with the data item only if said local existence means determines that no instance of the data item is present at the destination.

8. An apparatus as in claim 7, further comprising:

backup means for making copies of data items in the system, said backup means maintaining a backup record of identifiers of data items backed up, and invoking duplication means to copy only those data items whose data identifiers are not recorded in the backup record.

9. An apparatus as in claim 8, further comprising:

recovery means for retrieving a data item previously backed up by said backup means, based on the identifier of the data item, said recovery means using the backup record to identify the data item, and invoking access means to retrieve the data item.

10. An apparatus as in claim 2, wherein a location is a computer among a network of computers, the apparatus further comprising:

remote existence means for determining whether a data item is present at a remote location in the system from a current location in the system, based on the identifier of the data item, said remote location using local existence means at the remote location to determine whether the data item is present at the remote location, and providing the current location with an indication of the presence of the data item at the remote location.

11. An apparatus as in claim 4, wherein a location is a computer among a network of computers, the apparatus further comprising:

requesting means for requesting a data item at a current location in the system from a remote location in the system, based on the identifier of the data item, said remote location using access means at the remote location to obtain the data item and to send it to the current location if it is present.

12. An apparatus as in claim 1, further comprising:

context means for making and maintaining a context association between at least one contextual name of a data item in the system and the identifier of the data item; and

referencing means for obtaining the identifier of a data item in the system given a contextual name for the data item, using said context association.

13. An apparatus as in claim 12, further comprising:

assignment means for assigning a data item to a contextual name, invoking said identity means to determine the identifier of the data item, and invoking said context means to make or modify the context association between the contextual name of the data item and the identifier of the data item.

14. An apparatus as in claim 12, further comprising:

data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item;

access means for accessing a particular data item using the identifier of the particular data item; and

contextual name access means for accessing a data item in the system for a given context name of the data item, determining the data identifier associated with the given context name, and invoking said access means to access the data item using the data identifier.

15. An apparatus as in claim 11, further comprising:

transparent access means for accessing a data item from one of several locations, using the identifier of the data item, said transparent access means invoking said local existence means to determine if the particular data item is present at the current location, and, in the case when the particular data item is not present at the current location, invoking said requesting means to obtain the data item from a remote location.

16. An apparatus as in claim **15**, further comprising:

identifier copy means for copying an identifier of a data item from a source location to a destination location.

17. An apparatus as in claim **15**, further comprising:

context means for making and maintaining a context association between a contextual name of a data item in the system and the identifier of the data item;

context copy means for copying a data item from a source location to a destination location, given the contextual name of the data item, by copying only the context association between the contextual identifier and the data identifier from the source location to the destination location; and

transparent referencing means for obtaining a data item from one of several locations the system given a contextual name for the data item, said transparent referencing means invoking said context association to determine the data identifier of a data item given a contextual name, and invoking said transparent access means to access the data item from one of several locations given the identifier of the data item.

18. An apparatus as in claim **1**, wherein at least some of said data items are compound data items, each compound data item including at least some component data items in a fixed sequence, and wherein the identity means determines the identifier of a compound data item based on each component data item of the compound data item.

19. An apparatus as in claim **18**, wherein said compound data items are files and said component data items are segments, and wherein the identity means determines the identifier of a file based on the identifier of each data segment of the file.

20. An apparatus as in claim **18**, wherein said compound data items are directories and said component data items are files or subordinate directories, and wherein the identity means determines the identifier of a given directory based on each file and subordinate directory within the given directory.

21. An apparatus as in claim **11**, further comprising:

means for advertising a data item from a location in the system to at least one other location in the system, said means for advertising providing each of said at least one other location with the data identifier of the data item, and providing the data item to only those locations of said other locations that request said data item in response to said providing.

22. An apparatus as in claim **18**, further comprising:

local existence means for determining whether a particular data item is present at a particular location in the system, based on the identifier of the data item; and

compound copy means for copying a data item from a source to a destination in the data processing system, said compound copy means invoking said local existence means to determine whether the data item is present at the destination, and to determine, when the data item is a compound data item, whether the component data items of the compound data item are present at the destination, and providing said destination with the data item only if said local existence means determines that the data item is not present at the destination, and providing said destination with each component data item only if said local existence means determines that the component data item is not present at the destination.

23. An apparatus as in claim **11**, further comprising:

means for verifying the integrity of a data item obtained from the requesting means in response to providing the requesting with a particular data identifier, to confirm that the data item obtained from the requesting means is the same data item as the data item requested, the verifying means invoking the identity means to determine the data identifier of the obtained data item, and comparing the determined data identifier with the particular data identifier to verify the obtained data item.

24. An apparatus as in claim **2**, wherein a location is at least one of a storage location and a processing location, and wherein a storage location is at least one of a data storage device and a data storage volume, and wherein a processing location is at least one of a data processor and a computer.

25. An apparatus as in claim **3**, wherein at least some of said data items are compound data items, each compound data item including at least some component data items in a fixed sequence, and wherein the identity means determines the identifier of a compound data item based on the identifier of each component data item of the compound data item.

26. An apparatus as in claim **3**, further comprising:

context associating means for making and maintaining a context association, for any data item in the system, between the identifier of the data item and at least one contextual name of the data item at a particular location in the system;

means for obtaining the identifier of a data item in the system given a contextual name for the data item at a particular location in the system; and

logical copy means for associating the data identifier corresponding to a contextual name at a source location with a contextual name at a destination location in the data processing system.

27. An apparatus as in claim **25**, wherein said compound data items are files and said component data items are segments, and wherein the identity means determines the identifier of a file based on the identifier of each data segment of the file.

28. An apparatus as in claim **25**, further comprising:

compound copy means for copying a data item from a source location to a destination location in the data processing system, said compound copy means invoking said local existence means to determine whether the data item is present at the destination, and to determine, when the data item is a compound data item, whether the component data items of the compound data item are present at the destination, and providing said destination with the data item only if said local existence means determines that the data item is not present at the destination, and providing said destination with each component data item only if said local existence means determines that the component data item is not present at the destination.

29. An apparatus as in any of claims **1–28**, wherein a data item is at least one of a file, a database record, a message, a data segment, a data block, a directory, and an instance an object class.

30. A method of identifying a data item present in a data processing system for subsequent access to the data item, the method comprising:

determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

accessing a data item in the system using the identifier of the data item.

31. A method as in claim 30, further comprising:

making and maintaining, for a plurality of data items present in the system, an association between each of the data items and the identifier of each of the data items, wherein said accessing a data item accesses a data item via the association.

32. A method as in claim 31, further comprising:

assimilating a new data item into the system, by determining the identifier of the new data item and associating the new data item with its identifier.

33. A method for duplicating a given data item present at a source location to a destination location in a data processing system, the method comprising:

determining a substantially unique identifier for the given data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier;

determining, using the data identifier, whether the data item is present at the destination location; and

based on the determining whether the data item is present, providing the destination location with the data item only if the data item is not present at the destination.

34. A method as in claim 33, wherein the given data item is a compound data item having a plurality of component data items, the method further comprising:

for each data item of the component data items,

obtaining the component data identifier of the data item by determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier;

determining, using the obtained component data identifier, whether the data item is present at the destination; and

based on the determining, providing the destination with the data item only if the data item is not present at the destination.

35. A method for determining whether a particular data item is present in a data processing system, the method comprising:

(A) for each data item of a plurality of data items present in the system,

(i) determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

(ii) making and maintaining a set of identifiers of the plurality of data items; and

(B) for the particular data item,

(i) determining a particular substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

(ii) determining whether the particular identifier is in the set of data items.

36. A method of backing up, of a plurality of data items present in a data processing system, data items modified since a previous backup time in the data processing system, the method comprising:

(A) maintaining a backup record of identifiers of data items backed up at the previous backup time; and

(B) for each of the plurality of data items present in the data processing system,

(i) determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier;

(ii) determining those data items of the plurality of data items whose identifiers are not in the backup record; and

(iii) based on the determining, copying only those data items whose data identities are not recorded in the backup record.

37. A method as in claim 36, further comprising:

recording in the backup record the identifiers of those data items copied in said copying.

38. A method of locating a particular data item at a location in a data processing system, the method comprising:

(A) determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier;

(B) requesting the particular data item by sending the data identifier of the data item from the requester location to at least one location of a plurality of provider locations in the system; and

(C) on at least some of the provider locations,

(a) for each data item of a plurality of data items at the provider locations,

(i) determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only on the data in the data item, whereby two identical data items in the system will have the same identifier; and

(ii) making and maintaining a set of identifiers of data items,

(b) determining, based on the set of identifiers, whether the data item corresponding to the requested data identifier is present at the provider location; and

(c) based on the determining, when the provider location determines that the particular data item is present at the provider location, notifying the requestor that the provider has a copy of the given data item.

39. The method of claim 38, further comprising:

(a) for each data item of a plurality of data items present at said provider locations,

making and maintaining an association between the data item and the identifier of the data item,

(b) in response to said notifying, said client location copying said data item from one of said responding remote locations, using said association to access the data item given the data identifier.

40. A method of locating a particular data item among a plurality of locations, each of the locations having a plurality of data items, the method comprising:

determining, for the particular data item and for each data item of the plurality of data items, a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

determining the presence of the particular data item in each of the plurality of locations by determining whether the identifier of the particular data item is present at each of the locations.

41. The method of claim 30, wherein said accessing further comprises: for a given data identifier and for a given current location and a remote location in the system:

determining whether the data item corresponding to the given data identifier is present at the current location, and

based on said determining, if said data item is not present at the current location, fetching the data item from a remote location in the system to the current location.

42. The method of claim 41, further comprising:

for each contextual name at a location,

making and maintaining a context association between the context name of a data item and the identifier of said data item, and when some context association changes at said current location, and

notifying said remote location of a modification to the context association.

43. The method of claim 42, further comprising:

at said remote location, updating the association between the contextual identifier of the data item and the identifier of the data item.

44. The method of claim 43, further comprising:

from said remote location, notifying all other locations that said data item has been modified, by providing the contextual identifier and data identifier of said data item to said other locations.

45. The method of claim 44, further comprising, at each location notified that the data item has been modified:

modifying an association between the contextual identifier of the data item and the data identifier of the data item, to record that the data item has been modified.

46. A method of maintaining at least a predetermined number of copies of a given data item in a data processing system, at different locations in the data processing system, the data processing system being one wherein data is identified by a substantially unique identifier, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier, and wherein any data item in the system may be accessed using only the identifier of the data item, the method comprising:

(i) sending, from a first location in the system, the data identifier of the given data item to other locations in the system; and

(ii) in response to the sending, at each of the other locations,

(A) determining whether the data item corresponding to the data identifier is present at the other location, and based on the determining, and

(B) informing the first location whether the data item is present at the other location; and

(iii) in response to the informing from the other locations, at the first location,

(A) determining whether the data item is present in at least the predetermined number of other locations, and based on the determining,

(B) when less than the predetermined number of other locations have a copy of the data item, requesting some locations that do not have a copy of the data item make a copy of the data item.

47. A method as in claim 46, wherein said step (iii) further comprises:

(C) when more than the predetermined number of other locations have a copy of the data item present, requesting some locations that do have a copy of the data item present delete the copy of the data item.

48. A method as in any of claims 30–45, 46 and 47, wherein said data items are at least one of a file, a database record, a message, a data segment, a data block, a directory, and an instance of an object class.

* * * * *

CERTIFICATE OF SERVICE

I hereby certify that the foregoing CORRECTED BRIEF OF APPELLANT PERSONALWEB TECHNOLOGIES, LLC was served this 12th day of November, 2014 by operation of the Court's CM/ECF system per FED. R. APP. P. 25.

Date: November 12, 2014

/s/ Joel L. Thollander

CERTIFICATE OF COMPLIANCE

I hereby certify that the foregoing BRIEF OF APPELLANT PERSONALWEB TECHNOLOGIES, LLC:

1. complies with the type-volume limitation of FED. R. APP. P. 32(a)(7)(B), as extended by Order of this Court on October 15, 2014. This brief contains 18,499 words, excluding the parts of the brief exempted by FED. R. APP. P. 32(a)(7)(B)(iii) and FED. CIR. R. 32(b). Microsoft Word 2010 was used to calculate the word count.

2. complies with the typeface requirements of FED. R. APP. P. 32(a)(5) and the type style requirements of FED. R. APP. P. 32(a)(6). This brief has been prepared in a proportionally-spaced typeface using Microsoft Word 2010 in 14-point Times New Roman type style.

Date: November 7, 2014

/s/ Joel L. Thollander